

GCP Services



Cloud Storage

Google Cloud Storage is an online file storage service provided by Google as part of its cloud computing platform. It allows you to store and retrieve your data in the cloud, making it accessible from anywhere with an internet connection.



Compute Engine

Google Compute Engine is a cloud computing service that provides virtual machines for running applications and services. It allows you to easily create, configure, and manage virtual machines with various operating systems and hardware configurations.



BigQuery

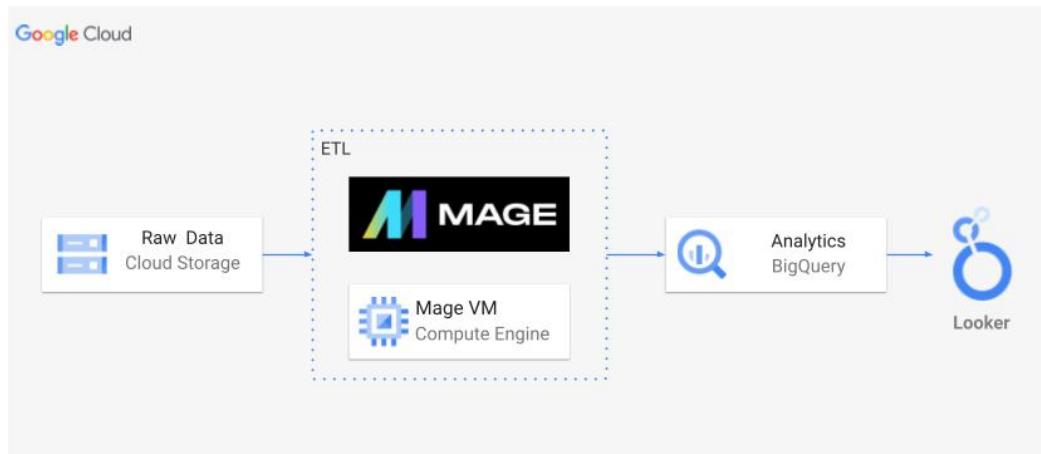
BigQuery is a cloud-based data warehouse provided by Google Cloud Platform that allows you to store, analyze, and query large datasets using SQL-like syntax. It is a serverless, highly scalable, and cost-effective solution that can process and analyze terabytes to petabytes of data in real-time.



Looker

Looker Studio is a web-based data visualization and reporting tool that allows you to create interactive dashboards and reports from a variety of data sources, including Google Analytics, Google Sheets, and BigQuery. It enables you to turn your data into informative and engaging visualizations, which can be easily shared and collaborated on with others.

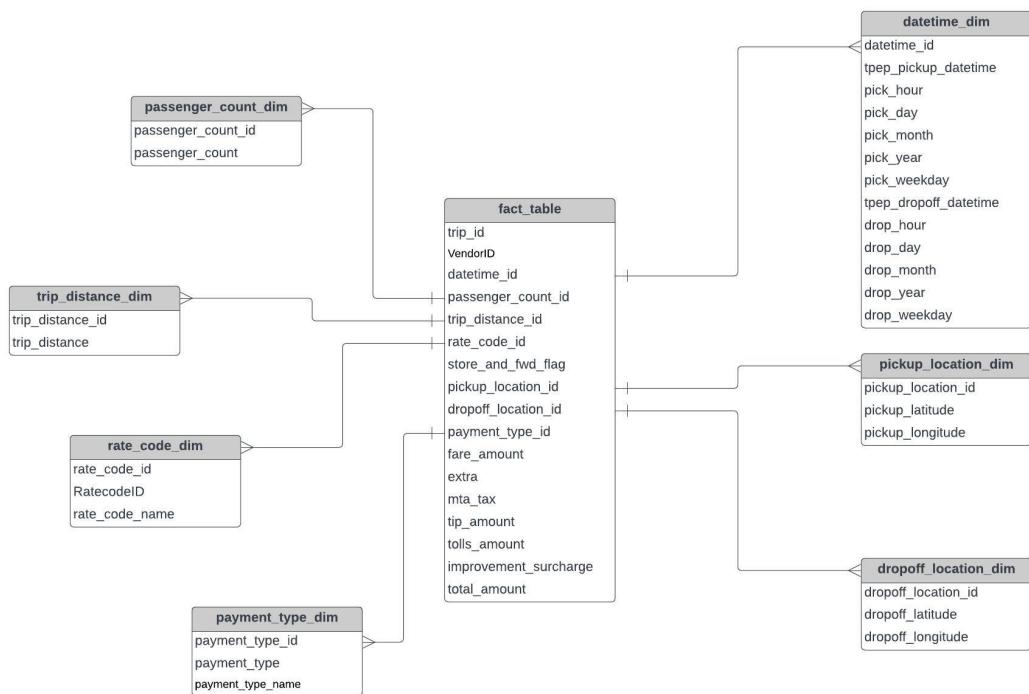
Arquitectura



Extraemos los datos de tráfico de la página <https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

The screenshot shows the NYC Taxi & Limousine Commission website. The main navigation bar includes links for About, Passengers, Drivers, Vehicles, Businesses, TLC Online, and a search bar. Below the navigation, there are four buttons: About TLC, Data and Reports, TLC Initiatives, and Contact TLC. The main content area is titled "TLC Trip Record Data". It contains a sidebar with links for Data, Pilot Programs, Reports, TLC Trip Record Data, Request Data, and a share/print button. The main content area describes the trip record data, noting it includes fields for pick-up and drop-off dates/times, locations, distances, fares, rates, payment types, and driver-reported passenger counts. It also mentions that the data is collected and provided to the NYC Taxicab & Limousine Commission by technology providers authorized under the TLC & Livery Passenger Enhancement Programs (TLEP/PEP). The data was not created by the TLC and is not guaranteed to be accurate or complete. A section titled "ATTENTION!" details changes to trip record files, including the switch to PARQUET format, monthly publication, additional columns, and new fields like "airport_fee".

Modelado de los datos



Se manipula el dataset inicial para poder formar la estructura de la tabla `fact_table`. La transformación se puede ver en el notebook del proyecto.

Fact Table																	Python	
...	trip_id	VendorID	datetime_id	passenger_count_id	trip_distance_id	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type_id	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improvement_surcharge	total_amount	
0	0	1	0	0	0	0	N	0	0	0	9.0	0.5	0.25	0.00	0.00	0.3		
1	1	1	1	1	1	1	N	1	1	1	11.0	0.5	0.5	3.05	0.00	0.3		
2	2	2	2	2	2	2	N	2	2	2	54.5	0.5	8.00	0.00	0.00	0.3		
3	3	2	3	3	3	3	N	3	3	3	31.5	0.0	0.5	3.78	5.54	0.3		
4	4	2	4	4	4	4	N	4	4	4	98.0	0.0	0.0	0.00	15.50	0.3		
...		
99995	99995	1	99995	99995	99995	99995	N	99995	99995	99995	5.0	0.0	0.5	0.00	0.00	0.3		
99996	99996	1	99996	99996	99996	99996	N	99996	99996	99996	14.0	0.0	0.5	2.00	0.00	0.3		
99997	99997	1	99997	99997	99997	99997	N	99997	99997	99997	29.0	0.0	0.5	8.80	5.54	0.3		
99998	99998	2	99998	99998	99998	99998	N	99998	99998	99998	5.5	0.5	0.5	1.36	0.00	0.3		
99999	99999	1	99999	99999	99999	99999	N	99999	99999	99999	6.0	0.0	0.5	0.00	0.00	0.3		

100000 rows × 17 columns

Iniciamos creando un bucket en el servicio Cloud Storage de GCP, y cargamos la data de Uber.

The screenshot shows the Google Cloud Platform Cloud Storage interface. The left sidebar has tabs for **Buckets**, **Supervisión**, and **Configuración**. The main area displays the **uber-mage-gcp-project** bucket details, including its location (us-east1), storage class (Standard), and access settings (No público). Below this, the **OBJETOS** tab is selected, showing a list of objects. One object, `uber_data.csv`, is listed with a size of 15.1 MB, type text/csv, and creation date of 20 sept 2023 16:27:10. The object is owned by Standard and last modified on 20 sept 2023 16:27:10. It has no public access and is encrypted by Administrada por Google.

Editamos los permisos del archivo para que cualquiera pueda acceder.

The screenshot shows the 'Edit control de acceso' dialog box over a Google Cloud Storage bucket details page. The dialog box has two options: 'Uniforme' (radio button) and 'Preciso' (radio button, selected). The 'Preciso' option is described as specifying access to individual objects via object-level permissions (LCA) in addition to bucket-level IAM permissions. A note below states that detailed access control overrides previous LCA configurations. Buttons at the bottom are 'CANCELAR' and 'GUARDAR'.

The screenshot shows the 'Acceso de edición' dialog box for the file 'uber_data.csv'. It displays a warning that the file is public and can be accessed from the internet. It includes fields for 'Entidad 1' (set to 'Public'), 'Nombre 1' (set to 'allUsers'), and 'Acceder 1' (set to 'Reader'). Buttons at the bottom are 'CANCELAR' and 'GUARDAR'.

The screenshot shows the 'Acceso público' section. It features a warning icon and the text 'Público para Internet' followed by a 'Copiar URL' link. Below this is a horizontal line.

Creamos una VM instance en el servicio Compute Engine de GCP para poder utilizar nuestra máquina virtual.

The screenshot shows the 'Create an instance' wizard. On the left, there's a sidebar with options like 'New VM instance', 'New VM instance from template', 'New VM instance from machine image', and 'Marketplace'. The main area has fields for 'Name' (uber-mage-gcp-instance-1), 'Region' (us-west1 (Oregon)), 'Zone' (us-west1-b), and 'Machine configuration' (General purpose tab selected). A table lists various machine types (C3, E2, N2, N2D, T2A, T2D, N1) with their descriptions, vCPUs, memory, and platforms. To the right, a 'Monthly estimate' table shows costs for 4 vCPU + 16 GB memory (\$97.84), 10 GB balanced persistent disk (\$1.00), and a total of \$98.84. There are tabs for 'Compute Engine pricing' and 'LESS'.

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

- Allow HTTP traffic
- Allow HTTPS traffic
- Allow Load Balancer Health Checks

Se puede ver la instancia creada. Hacemos click en la opción de SSH para conectarnos.

The screenshot shows the 'Compute Engine' dashboard under 'Virtual machines'. The 'VM instances' section lists 'uber-mage-gcp-instance-1' with details like status (Running), name, zone (us-east1-b), internal IP (10.142.0.2), external IP (35.231.39.159), and a 'Connect' button. Below is a 'Related actions' section with a 'SHOW' button. At the bottom, a terminal window titled 'SSH-in-browser' shows a successful connection to the instance, displaying the Debian prompt 'Linux uber-mage-gcp-instance-1 5.10.0-25-cloud-amd64 #1 SMP Debian 5.10.191-1 (2023-08-16) x86_64'. The terminal also shows the standard Debian copyright notice and a prompt for further commands.

Al ingresar a la consola, corremos los siguientes comandos para poder utilizar python3.

```
sudo apt-get update -y
sudo apt-get install python3-distutils
sudo apt-get install python3-apt
wget https://bootstrap.pypa.io/get-pip.py
sudo python3 get-pip.py
```

Luego, instalamos las dependencias.

```
sudo pip3 install pandas
sudo pip3 install google-cloud
sudo pip3 install google-cloud-bigquery
sudo pip3 install mage-ai (Entorno para manejar el workflow)
```

Empezamos a utilizar **mage**.

```
mage start uber_gcp_project
```

```
salazar@jhoncloud:~/uber-mage-gcp-instance-1:~$ mage start uber_gcp_project
WARNING:traitlets:Message signing is disabled. This is insecure and not recommended!
WARNING:traitlets:Message signing is disabled. This is insecure and not recommended!
INFO:mage ai.server:Switch active kernel; python3
INFO:[alembic.runtime.migration] Context impl SQLiteImpl.
INFO:[alembic.runtime.migration] Will assume non-transactional DDL.
INFO:[alembic.runtime.migration] Running upgrade > c07a23ff7f82b, Add initial tables.
INFO:[alembic.runtime.migration] Running upgrade c07a23ff7f82b -> 52ab80005742, Add variables to pipeline schedule.
INFO:[alembic.runtime.migration] Running upgrade 52ab80005742 -> 053ee2c10d85, Add completed_at.
INFO:[alembic.runtime.migration] Running upgrade 053ee2c10d85 -> 3fafdf217efaf7, Add event matcher model.
INFO:[alembic.runtime.migration] Running upgrade 3fafdf217efaf7 -> 067326f43bc3, Add variables to PipelineRun.
INFO:[alembic.runtime.migration] Running upgrade 067326f43bc3 -> b01be687e537, Add started_at to block run.
INFO:[alembic.runtime.migration] Running upgrade b01be687e537 -> 84de4cc4cd6126, Add sla to pipeline schedule.
INFO:[alembic.runtime.migration] Running upgrade 84de4cc4cd6126 -> 5cd59ec4cf1d, Add passed_sla to pipeline run.
INFO:[alembic.runtime.migration] Running upgrade 5cd59ec4cf1d -> 2266370f5898, Add indexes to pipeline run.
INFO:[alembic.runtime.migration] Running upgrade 2266370f5898 -> 8971d4cd5b39, add event_variables and metadata to pipeline_runs
INFO:[alembic.runtime.migration] Running upgrade 8971d4cd5b39 -> e5cd5f57a1c60, Add metrics to block runs.
INFO:[alembic.runtime.migration] Running upgrade e5cd5f57a1c60 -> 7ac6fed06918, Add token column to pipeline_schedule table.
INFO:[alembic.runtime.migration] Running upgrade 7ac6fed06918 -> 6aeccc9bc451c, Update schedule type enum.
INFO:[alembic.runtime.migration] Running upgrade 6aeccc9bc451c -> 26305e46d5f2, Create backfills table.
INFO:[alembic.runtime.migration] Running upgrade 26305e46d5f2 -> 1d9e65aeeffdd, Add settings to pipeline schedule.
INFO:[alembic.runtime.migration] Running upgrade 1d9e65aeeffdd -> 97ff9f5f5a3c0, Create users, oauth2 applications, and oauth2 access tokens.
INFO:[alembic.runtime.migration] Running upgrade 97ff9f5f5a3c0 -> 643b6e65e014, Add unique indexes on authentication tables.
Checking port 6789...
INFO:mage ai.server:Server Mage is running at http://localhost:6789 and serving project /home/salazarjhoncloud/uber_gcp_project
INFO:mage ai.server:Initializing block cache.
INFO:mage ai.server:Initializing tag cache.
INFO [alembic.runtime.migration] Running upgrade 643b6e65e814 -> 1f9353eddbbc6, Add secrets table
```

Vamos a la sección de interfaces de red de nuestra instancia. Seleccionamos la interfaz *nic0* para agregar una regla de firewall que nos permite utilizar el puerto 6789 de mage.

Network interfaces

Name	Network	Subnetwork	Primary internal IP address	Alias IP ranges	IP stack type	External IP address	Network Service Tier
nic0	default	default	10.142.0.2	—	IPv4	35.231.39.159 (Ephemeral)	Premium

En el menú vertical seleccionamos la opción **Firewall**.

The screenshot shows the Google Cloud VPC network interface details for the 'uber-mage-gcp-project' instance. In the left sidebar, 'Firewall' is selected under the 'VPC network' category. The main pane displays the 'Selected network interface: nic0'. Under 'Network interface details', there is a table with one row for 'nic0'. Under 'VM instance details', there is a table with one row for the VM instance. Under 'Firewall and routes details', there is a table titled 'FIREWALLS' with one rule named 'vpc-firewall-rules'. The rule has an enforcement order of 1, is a VPC firewall rule, and is global.

En este punto seleccionamos la opción de crear una nueva regla de firewall.

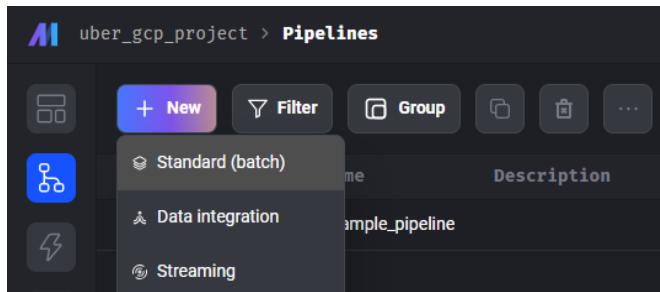
La configuración es la siguiente.

Network Security	Network default
Secure Web Proxy	Priority * 1000 <small>CHECK PRIORITY OF OTHER FIREWALL RULES</small>
Cloud Armor	Direction Ingress
Cloud Armor policies	Action on match Allow
Adaptive Protection	Targets All instances in the network
Managed Protection	Source filter IPv4 ranges
Cloud IDS	Source IPv4 ranges * 0.0.0.0/0
IDS Dashboard	Second source filter None
IDS Endpoints	Destination filter None
IDS Threats	Protocols and ports <input checked="" type="radio"/> TCP Ports 6789
Cloud Firewall	E.g. 20, 50-60
Firewall policies	
Threats PREVIEW	
Firewall endpoints PREVIEW	
Common components	
Security profiles PREVIEW	
SSL policies	

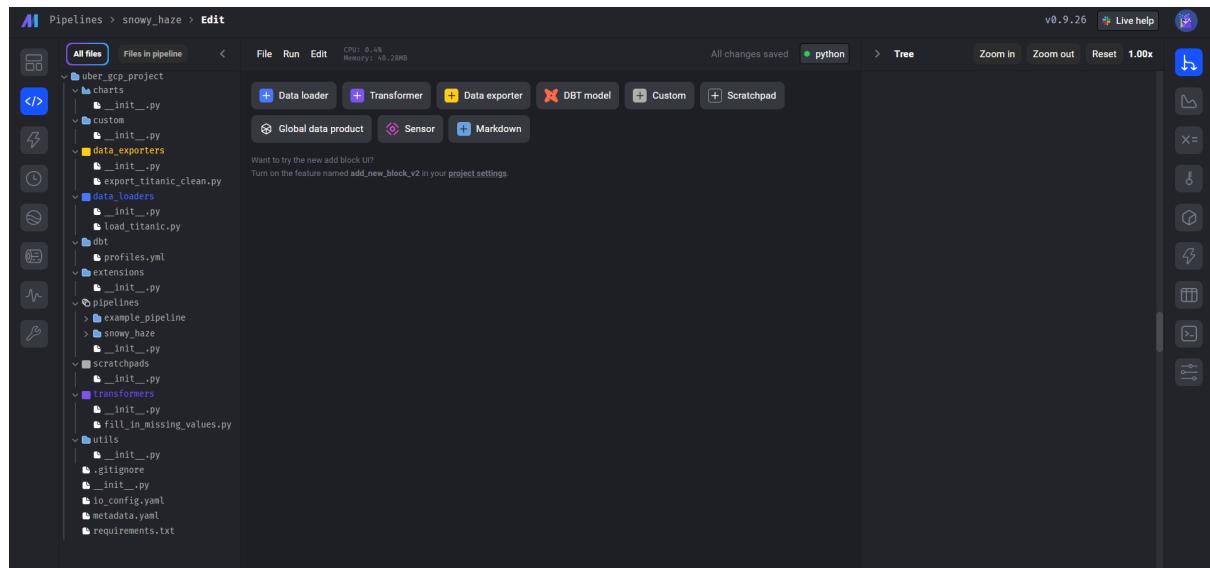
Al intentar acceder nuevamente a **mage**, tomando la dirección IP externa de la instancia y el puerto proporcionado por mage, podemos ver la siguiente interface.

Status	Name	Description	Type	Updated at	Created at	Tags	Blocks	Triggers
no schedules	example_pipeline		Standard	-	-	none	3	0

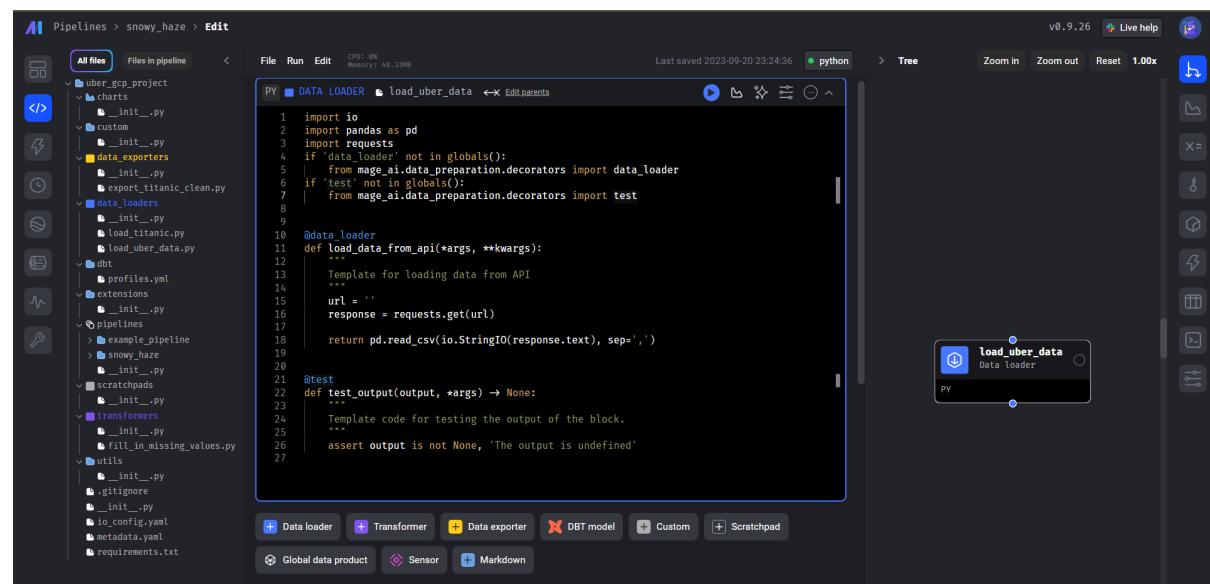
Creamos un escenario Standard.



Mage ofrece template para apoyarse y generar scripts más fácilmente.

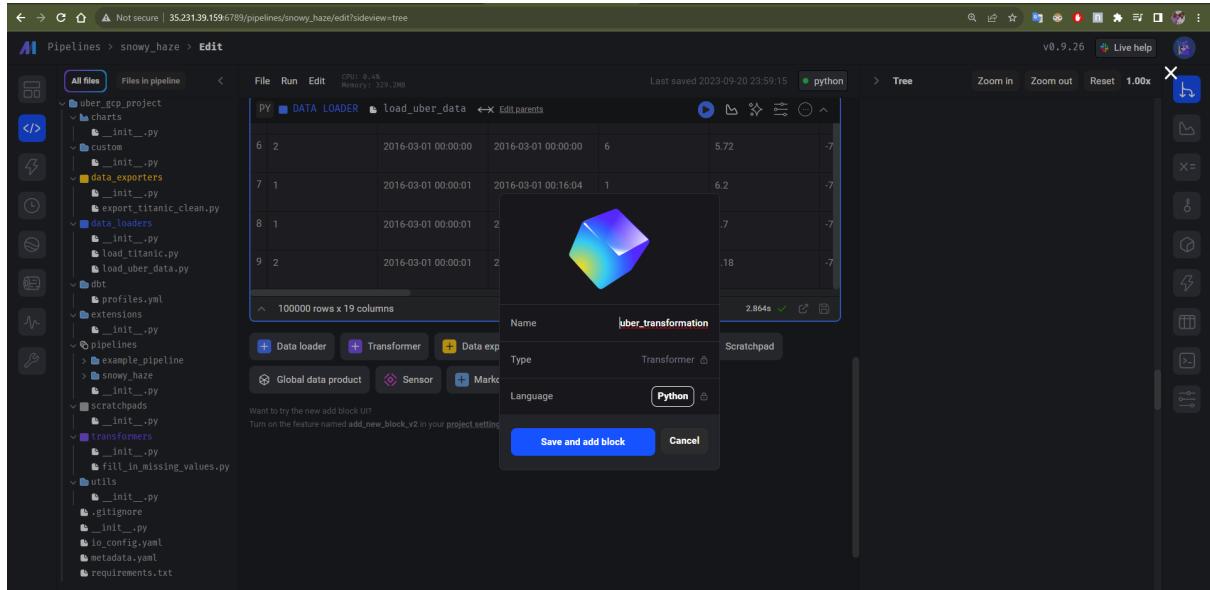


Creamos un Data loader de tipo API, ya que podemos acceder a nuestro storage de GCP de forma sencilla a través de la url pública.



Corremos este bloque y podemos ver los datos del csv de uber.

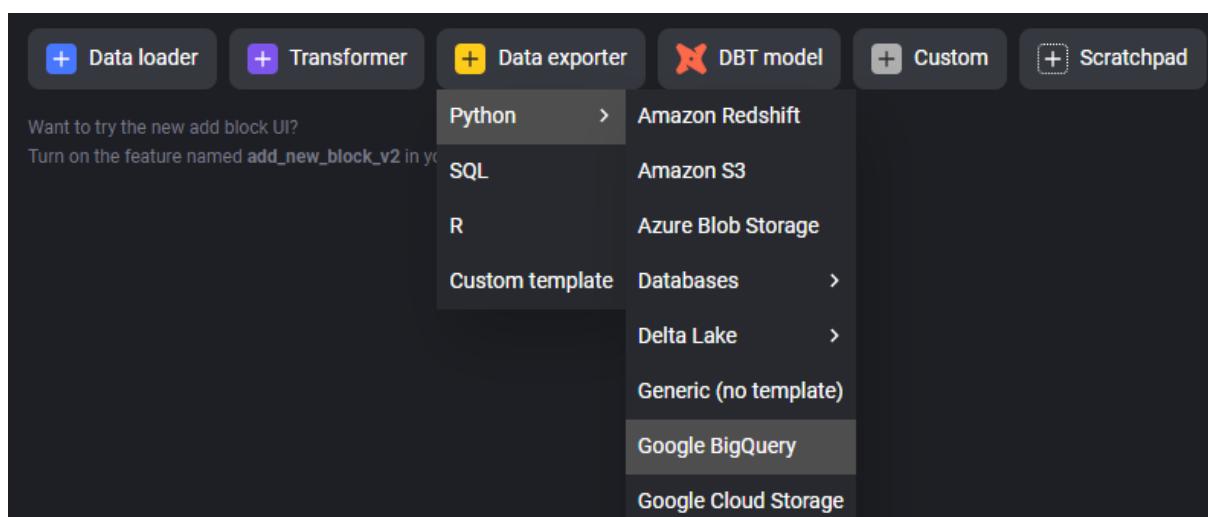
Luego creamos un Transformer (Generic) para manipular estos datos y obtener el modelo mencionado anteriormente.



En el transformer simplemente pegamos el código realizado en el notebook. Hasta el momento tenemos estos 2 bloques. Ahora cargaremos la información en BigQuery.



Creamos el bloque Data exporter.



Para poder conectarnos desde la VM a cualquier servicio de GCP necesitamos unas credenciales. Vamos a la sección de APIs & Services y seleccionamos la opción Credentials.

The screenshot shows the Google Cloud Platform interface for managing APIs & Services. On the left, a sidebar menu has 'Enabled APIs & services' selected. The main area displays traffic metrics: 6/s for 'Traffic' and 4/s for 'Page usage agreements'. A search bar and filter icons are at the top of the traffic table.

Seleccionamos crear credenciales y elegimos Service account.

The screenshot shows the 'Credentials' creation page. Under 'API key', it says 'Identifies your project using a simple API key to check quota and access'. Under 'OAuth client ID', it says 'Requests user consent so your app can access the user's data'. Under 'Service account', it says 'Enables server-to-server, app-level authentication using robot accounts'. A 'Remember me' checkbox is checked. Below these options is a 'Help me choose' section with a 'Name' input field and a note 'No API keys to display'.

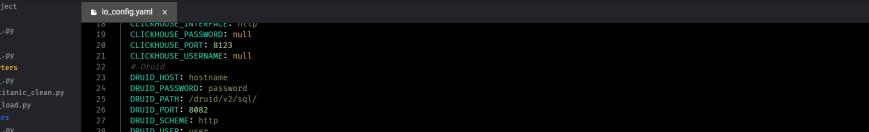
Escogemos el rol de BigQueryAdmin para poder crear tablas, editar, leer, etc.

This is a step in the service account creation process. It asks for permission to grant the service account access to the project. It includes a dropdown for selecting a role ('BigQuery Admin'), an optional IAM condition section, and a 'CONTINUE' button.

Seleccionamos la cuenta creada y generamos una key en formato JSON.

The screenshot shows the 'IAM & Admin' section of the Google Cloud console. In the 'Service Accounts' tab, under 'Keys', a modal dialog is open titled 'Create private key for "uber_dataengineering_project"'. It explains that the key can pose a security risk if compromised and recommends using Workload Identity Federation. It provides options for 'Key type': 'JSON' (selected) and 'P12'. At the bottom are 'CANCEL' and 'CREATE' buttons.

Nos dirigimos al archivo `io_config.yaml` e ingresamos nuestras credenciales generadas para la sección `GOOGLE_SERVICE_ACC_KEY`.



```
#!/usr/bin/python
# This file contains configuration for the ClickHouse pipeline.
# It defines the connection parameters for ClickHouse, Druid, and Google Cloud Storage.

# ClickHouse connection
clickhouse_host = "localhost"
clickhouse_port = 9102
clickhouse_username = "default"

# Druid connection
druid_host = "localhost"
druid_password = "password"
druid_path = "/druid/v2/sql/"
druid_port = 8882
druid_scheme = "http"
druid_user = "user"
druid_tz = "UTC"

# Database connection
duckdb_database = "database"
duckdb_schema = "main"

# Google Cloud Storage connection
google_service_account_key_file = "/path/to/your/service/account/key.json"

# MongoDB connection
mongod_host = "localhost"
mongod_port = 27017
mongod_user = "user"
mongod_password = "password"
mongod_dbname = "database"
mongod_collection = "collection"
mongod_ssl = False
```

Ingresamos al servicio BigQuery (Datawarehouse) y creamos nuestro dataset dentro del proyecto correspondiente.

The screenshot shows the Google Cloud SQL workspace. On the left, there's a sidebar with various icons and a search bar. The main area has a heading "Welcome to your SQL Workspace!" and a "Get started" section with a "COMPOSE A NEW QUERY" button. Below that is a "Try with sample data" section featuring a "Google Trends Demo Query" with a "VIEW DATASET" button. Further down is an "Add your own data" section with options for "Local file" (with a "LAUNCH THIS GUIDE" button), "Google Drive" (with a "LAUNCH THIS GUIDE" button), and "Google Cloud Storage" (with a "LAUNCH THIS GUIDE" button). On the right, a "Create dataset" dialog is open, containing fields for "Project ID" (set to "uber-image-gcp-project"), "Dataset ID" (set to "uber_image_dataset"), and "Location type" (set to "Region" with "US" selected). There are also sections for "Default table expiration" (unchecked) and "Advanced options". At the bottom are "CREATE DATASET" and "CANCEL" buttons.

Welcome to your SQL Workspace!

Get started

COMPOSE A NEW QUERY

TRY WITH SAMPLE DATA

Try the Google Trends Demo Query

VIEW DATASET

ADD YOUR OWN DATA

Local file

Upload a local file

LAUNCH THIS GUIDE

Google Drive

Google storage service

LAUNCH THIS GUIDE

Google Cloud Storage

Google object storage service

LAUNCH THIS GUIDE

Create dataset

Project ID
uber-image-gcp-project

Dataset ID *
uber_image_dataset

Letters, numbers, and underscores allowed

Location type *

Region
Specify a region to colocate your datasets with other Google Cloud services.

Multi-region
Allow BigQuery to select a region within a group to achieve higher quota limits.

Multi-region *
US (multiple regions in United States)

Default table expiration

Enable table expiration

Default maximum table age

Advanced options

CREATE DATASET CANCEL

En el bloque de Data Exporter ingreamos los datos de nuestro dataset en BigQuery y lo ejecutamos. Como se puede ver, la conexión fue exitosa y se creó la nueva tabla.

The screenshot shows the Mage AI Pipeline interface with the following components:

- Pipeline Structure:** On the left, a tree view shows the pipeline structure with files like `uber_gcp_project`, `charts`, `custom`, `data_exporters`, `data_loaders`, `dbt`, `extensions`, `pipelines`, `scratches`, and `transformers`.
- Code Editor:** The central area displays Python code for a `DATA EXPORTER` named `uber_bg_load`. The code imports `BigQuery` and `ConfigFileLoader` from `mage_ai.io.bq`, and `DataFrame` from `pandas`. It defines a function `def export_data_to_big_query(data, **kwargs)` which uses a template from `io.config.yaml` to export data to a BigQuery warehouse.
- DAG Diagram:** To the right, a Directed Acyclic Graph (DAG) is shown. It consists of three nodes:
 - `load_uber_data`: A `Data loader` node (blue icon).
 - `uber_transformation`: A `Transformer` node (purple icon).
 - `uber_bg_load`: A `Data exporter` node (yellow icon).The nodes are connected sequentially: `load_uber_data` leads to `uber_transformation`, which then leads to `uber_bg_load`.
- Terminal Output:** At the bottom, a terminal window shows the command `python uber_bg_load` being run, with output indicating the BigQuery connection and data export process.

The screenshot shows the Google Cloud BigQuery interface. In the top navigation bar, it says "Google Cloud" and "uber-mage-gcp-project". A search bar at the top right contains the placeholder "Buscar () recursos, documentos, productos y más" and a "Buscar" button. Below the search bar is a toolbar with icons for file operations like "AGREGAR", "COMPARTIR", "COPIAR", "INSTANTÁNEA", "BORRAR", and "EXPORTAR".

The main area is titled "Explorador" and shows a tree view of resources under "uber-mage-gcp-project" and "uber_mage_dataset". A specific table named "fact_table" is selected. At the top of the table view, there are tabs for "ESQUEMA", "DETALLES", "VISTA PREVIA", "LINAJE", "PERFIL DE DATOS", and "CALIDAD DE LOS DATOS".

The "ESQUEMA" tab displays the schema of the "fact_table" with the following columns:

Nombre del campo	Tipo	Modo	Clave	Intercalación	Valor predeterminado	Etiquetas de políticas	Descripción
trip_id	INTEGER	NULLABLE					
VendorID	INTEGER	NULLABLE					
datetime_id	INTEGER	NULLABLE					
passenger_count_id	INTEGER	NULLABLE					
trip_distance_id	INTEGER	NULLABLE					
rate_code_id	INTEGER	NULLABLE					
store_and_fwd_flag	STRING	NULLABLE					
pickup_location_id	INTEGER	NULLABLE					
dropoff_location_id	INTEGER	NULLABLE					
payment_type_id	INTEGER	NULLABLE					
fare_amount	FLOAT	NULLABLE					
extra	FLOAT	NULLABLE					
mta_tax	FLOAT	NULLABLE					
tip_amount	FLOAT	NULLABLE					
tolls_amount	FLOAT	NULLABLE					
improvement_surcharge	FLOAT	NULLABLE					

At the bottom of the table view, there are buttons for "EDITAR ESQUEMA" and "VER POLÍTICAS DE ACCESO DE FILA".

Ejecutamos nuevamente este proceso pero para todas las tablas.

The screenshot shows the Alteryx Designer interface. On the left, the "Pipelines > snowy_haze > Edit" sidebar lists various components and files. The main workspace shows a Python script named "DATA EXPORTER" with the following code:

```

PY DATA EXPORTER  uber_bq_load ↔ 1 parent
for key, value in data.items():
    table_id = f'{uber_mage_gcp_project}.uber_mage_dataset.{key}'.format(key)
    BigQuery.with_config(ConfigFileLoader(config_path, config_profile)).export(
        Dataframe(value),
        table_id,
        if_exists='replace', # Specify resolution policy if table name already exists
    )

```

The status bar indicates "Saving changes..." and "python". To the right, a "Tree" view shows the data flow:

- "load_uber_data" (Data loader) feeds into "uber_transformation" (Transformer).
- "uber_transformation" feeds into "uber_bq_load" (Data exporter).

This screenshot shows the Google Cloud BigQuery interface, similar to the one above, but with a different dataset selected. It displays the schema of the "fact_table" with the same 15 columns as the previous screenshot.

Ejecutamos una sentencia sencilla.

The screenshot shows the BigQuery interface. On the left, the 'Explorador' sidebar lists datasets and tables under 'uber-mage-gcp-project'. A search bar at the top says 'Comienza a escribir para buscar'. The main area shows a query in the editor: 'SELECT * FROM uber-mage-gcp-project.uber_mage_dataset.fact_table LIMIT 10;'. Below the editor is a table titled 'Resultados de la consulta' with 10 rows of data from the fact_table.

Fila	trip_id	VendorID	datetime_id	passenger_count_id	trip_distance_id	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type_id
1	354	2	354	354	354	354	N	354	354	354
2	1309	2	1309	1309	1309	1309	N	1309	1309	1309
3	2560	2	2560	2560	2560	2560	N	2560	2560	2560
4	4036	2	4036	4036	4036	4036	N	4036	4036	4036
5	5196	2	5196	5196	5196	5196	N	5196	5196	5196
6	5462	2	5462	5462	5462	5462	N	5462	5462	5462
7	6465	2	6465	6465	6465	6465	N	6465	6465	6465
8	7555	2	7555	7555	7555	7555	N	7555	7555	7555
9	8415	2	8415	8415	8415	8415	N	8415	8415	8415
10	10010	2	10010	10010	10010	10010	N	10010	10010	10010

En este punto, queremos formar una nueva tabla con columnas provenientes de todas las tablas creadas anteriormente. El query final es el siguiente.

This screenshot shows a more complex query being created. The editor contains a CREATE OR REPLACE TABLE statement to create a new table 'data-with-dashai.uber_data_engineering.yt_tb1_analytics' from various source tables. The resulting table structure is shown in the results table below, which includes columns like trip_id, VendorID, tpep_pickup_datetime, tpep_dropoff_datetime, passenger_count, trip_distance, rate_code_name, pickup_latitude, pickup_longitude, dropoff_latitude, dropoff_longitude, and payment_type_name.

Fila	trip_id	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	rate_code_name	pickup_latitude	pickup_longitude	dropoff_latitude	dropoff_longitude	payment_type_name
1	1	1	2016-03-01T00:00:00	2016-03-01T00:07:55	1	2.5	Standard rate	40.7651519773...	-73.976456054...	40.7461208227...	-74.0042648515...	Credit card
2	1	1	2016-03-01T00:00:00	2016-03-01T00:11:06	1	2.9	Standard rate	40.7679252645...	-73.983423608...	40.7316167409...	-74.009432983...	Credit card
3	2	2	2016-03-01T00:00:00	2016-03-01T00:31:06	2	19.98	Standard rate	40.6448057220...	-73.8820205688...	40.67576680590...	-73.745407104...	Credit card
4	3	2	2016-03-01T00:00:00	2016-03-01T00:00:00	3	10.78	Standard rate	40.7698135379...	-73.8834185791...	40.75776672563...	-73.9696520865...	Credit card
5	4	2	2016-03-01T00:00:00	2016-03-01T00:00:00	5	30.43	Newark	40.7918252922...	-73.9717407226...	40.6595510058...	-74.1771679798...	Credit card
6	5	2	2016-03-01T00:00:00	2016-03-01T00:00:00	5	5.92	Standard rate	40.7053833007...	-74.0171966552...	40.7578689575...	-73.9780761201...	Credit card
7	6	2	2016-03-01T00:00:00	2016-03-01T00:00:00	6	5.72	Standard rate	40.7278405297...	-73.9945831298...	0.0	0.0	Cash

Creamos la tabla y queda almacenada en nuestro proyecto. Es la que usaremos para armar el dashboard. Para ellos ingresamos a Looker Studio y creamos un nuevo informe vacío.

The screenshot shows the Looker Studio interface. The top navigation bar has tabs for 'Crear', 'Recientes', 'Informes', 'Fuentes de datos', and 'Explorador'. The 'Recientes' tab is selected. Below it, there's a section for 'Empezar con una plantilla' with options like 'Informe vacío Looker Studio', 'Informe "Tutorial" Looker Studio', 'Acme Marketing Google Analytics', 'Search Console Report Search Console', 'Google Ads Overview Google Ads', and 'YouTube Channel Report YouTube Analytics'. At the bottom, there's a search bar for 'Nombre', a dropdown for 'Quienquiera es el propietario', a dropdown for 'Abierto última vez por mi...', a dropdown for 'Ubicación', and a 'Crear un informe' button with a bar chart icon.

Crear un informe
Use el botón Crear para añadir uno.

Elegimos la opción de conectarnos a BigQuery.



Escogemos nuestra tabla.

PROYECTOS RECIENTES	Proyecto	Conjunto de datos	Tabla
MIS PROYECTOS	Introduzca el ID del proyecto manualmente	uber_mage_dataset	datetime_dim dropoff_location_dim fact_table passenger_count_dim payment_type_dim pickup_location_dim rate_code_dim tbl_analytics trip_distance_dim
PROYECTOS COMPARTIDOS	uber-mage-gcp-project		
CONSULTA PERSONALIZADA	My First Project		
CONJUNTOS DE DATOS PÚBLICOS			

Dentro del informe podemos manejar diferentes elementos para ilustrar nuestra información de la mejor manera. En principio agregamos filtros, luego unas cifras relevantes, un mapa con las rutas de los viajes realizados y un par de gráficos de barras.

Para agregar el mapa, debemos crear un nuevo campo basado en la latitud y longitud del pickup.

Añadir un campo

Añadir un parámetro

Nombre del campo: pick_location

Fórmula: 1 CONCAT(pickup_latitude , " , pickup_longitude)

Nombre: pick_location

Tipo de datos:

- 123 Numérico
- RBC Texto
- Fecha y hora
- Booleano
- Información geográfica
- Moneda
- URL
- Personalizado

Continentes:

- Continente
- Subcontinente
- País
- Subdivisión del país (primer nivel)
- Subdivisión del país (segundo nivel)
- Área de mercado designada
- Ciudad
- Código postal
- Dirección
- Latitud, longitud

Finalmente el dashboard nos queda de la siguiente forma.

Uber Dashboard

Archivo Editar Vista Insertar Página Organizar Recurso Ayuda

Añadir página Añadir datos Añadir un gráfico Añadir un control Tema y diseño Pausar actualizaciones

Filters

Total Revenue: 1,6 M Record Count: 100,0 mil Avg Trip Distance: 3,03 Avg Fare Amount: 13,25 Avg Tip Amount: 1,87

Map

Mapa de Nueva York con puntos de viaje coloridos. Se observan una gran cantidad de viajes en el centro de la ciudad, especialmente en el distrito financiero.

Summary

Fecha de la última actualización: 21/6/2023 15:08:49

Chats

Gráfico de barras que muestra la distribución de los tipos de pagos:

Tipo de pago	Cantidad
Taxi or ride-hailing	~100
Bank	~40
Cash	~30
Credit card	~15
Tip charge	~10
Drop-off	~5

Gráfico de barras que muestra la distribución de los tipos de tarifas:

Tipo de tarifa	Cantidad
Standard	~100
Surge	~40
Peak surge	~30
Off-peak	~10

Datos

Búsquedas

tbl_analytics

- dropoff_latitude
- dropoff_longitude
- extra
- fare_amount
- improvement_surcharge
- mta_tax
- passenger_count
- payment_type_name
- pickup_location
- pickup_latitude
- pickup_longitude
- rate_code_name
- tip_amount
- total_amount
- trip_distance
- trip_id
- VendorID
- Record Count

Añadir un campo Añadir un parámetro Añadir datos