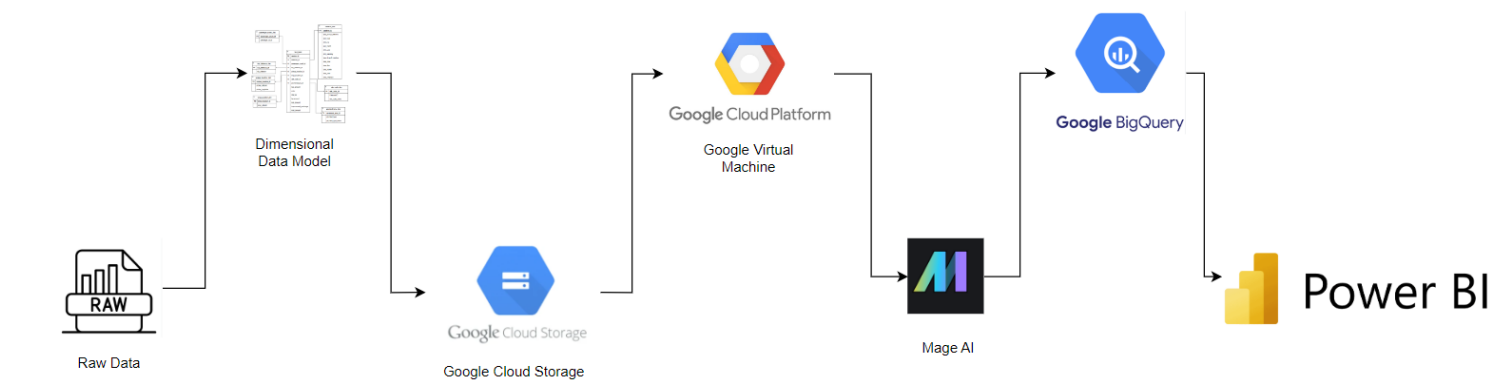


GCP DATA PIPELINE



Here I present the end-to-end Google Data Pipeline which I created to analyse the Uber data and visualize it on a Power BI Dashboard.

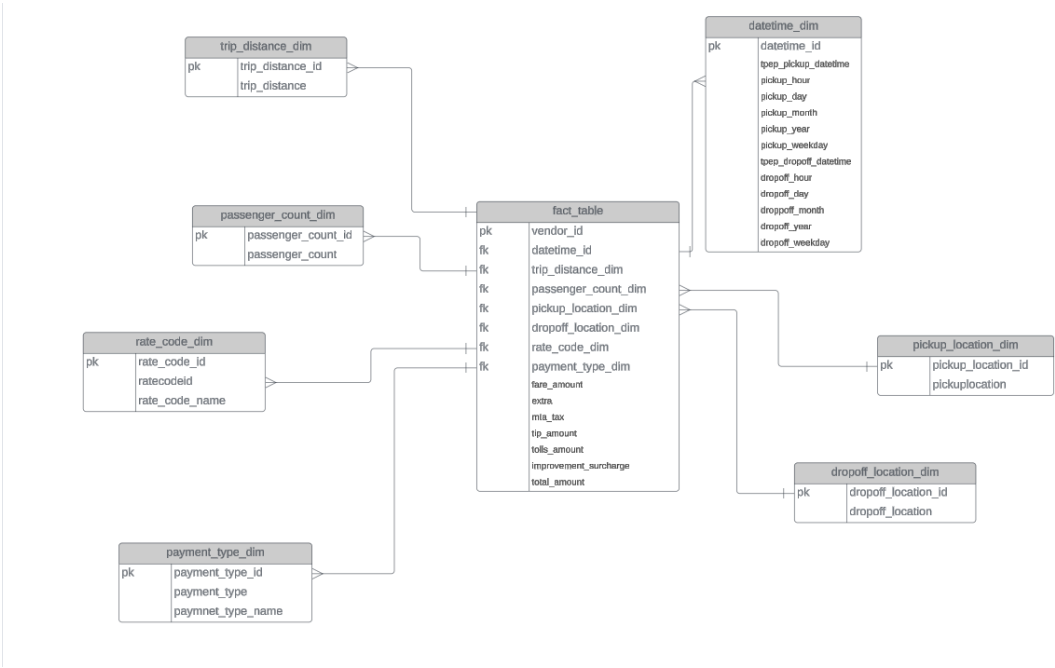
- The resources and Tools I used are -
- Raw data –Acquired the Raw data through open source portals
 - Draw.io-Used Draw.io to perform Dimensional modelling of the data
 - Google Cloud Storage –To store the raw
 - Google Virtual Machine-To gain the computing efficiency
 - Mage AI – To create data pipeline
 - Google BigQuery- To Query the data and make relations
 - Power BI-To make visualization on the insights gathered on BigQuery

Let’s first explore the data which we have

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	RatecodeID	store_and_fwd_flag
0	1	2016-03-01	2016-03-01 00:07:55	1	2.50	-73.976746	40.765152	1	N
1	1	2016-03-01	2016-03-01 00:11:06	1	2.90	-73.983482	40.767925	1	N
2	2	2016-03-01	2016-03-01 00:31:06	2	19.98	-73.782021	40.644810	1	N
3	2	2016-03-01	2016-03-01 00:00:00	3	10.78	-73.863419	40.769814	1	N
4	2	2016-03-01	2016-03-01 00:00:00	5	30.43	-73.971741	40.792183	3	N

Based on the Data I had I created a Dimensional Model of the data on Draw.io .

Here is the Dimensional Model I created

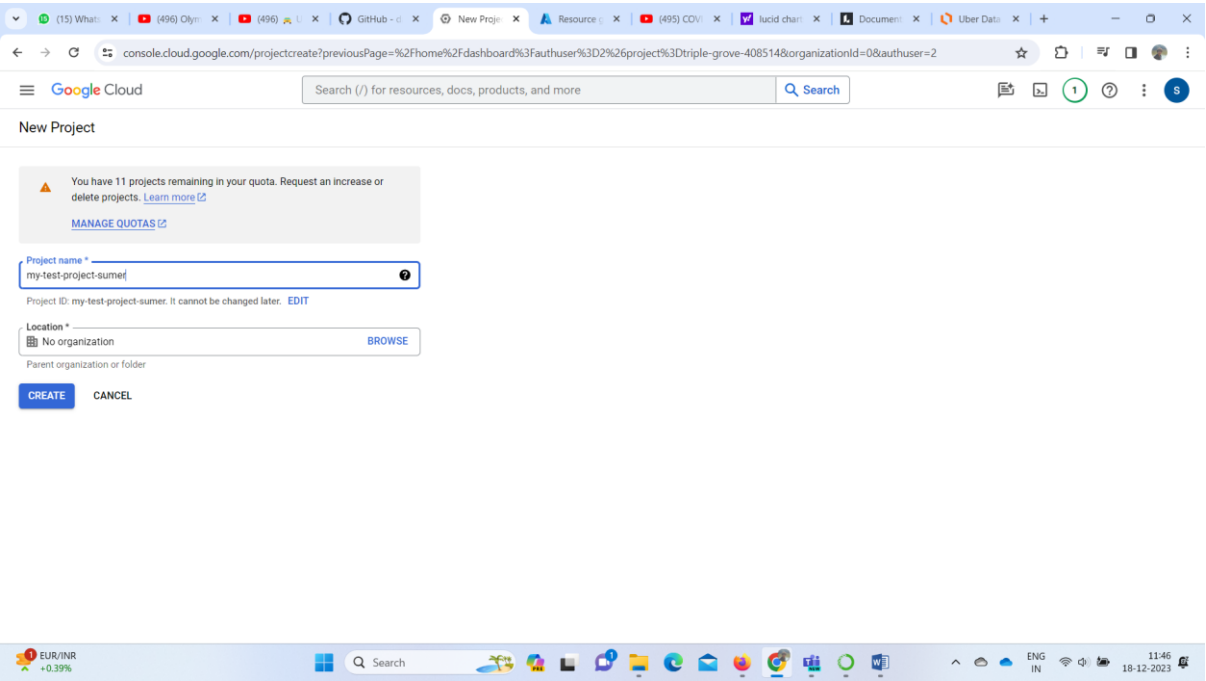


The dimension model (Star Schema) had

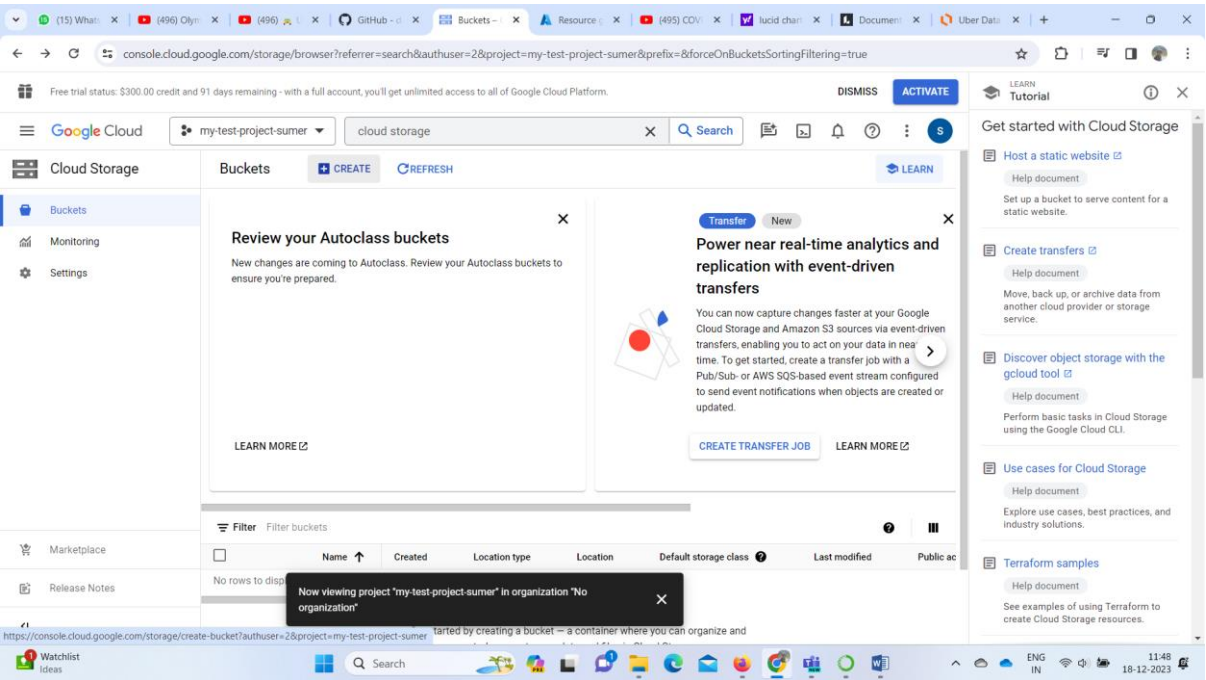
1. fact table
2. pickup location table
3. dropoff location table
4. passenger count
- 5.trip distance
- 6.date time
- 7.rate code type

8.payment type

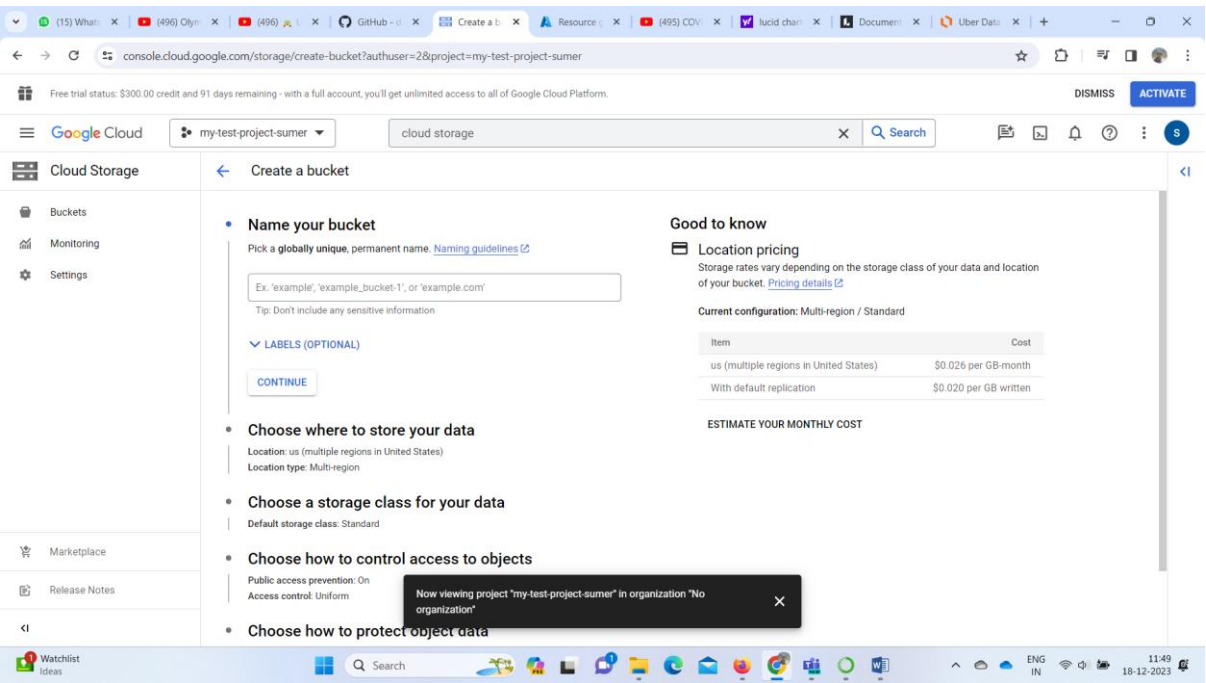
Once we had made our Star schema Data warehouse ,I was ready to login to my GCP console. The first step was create a project

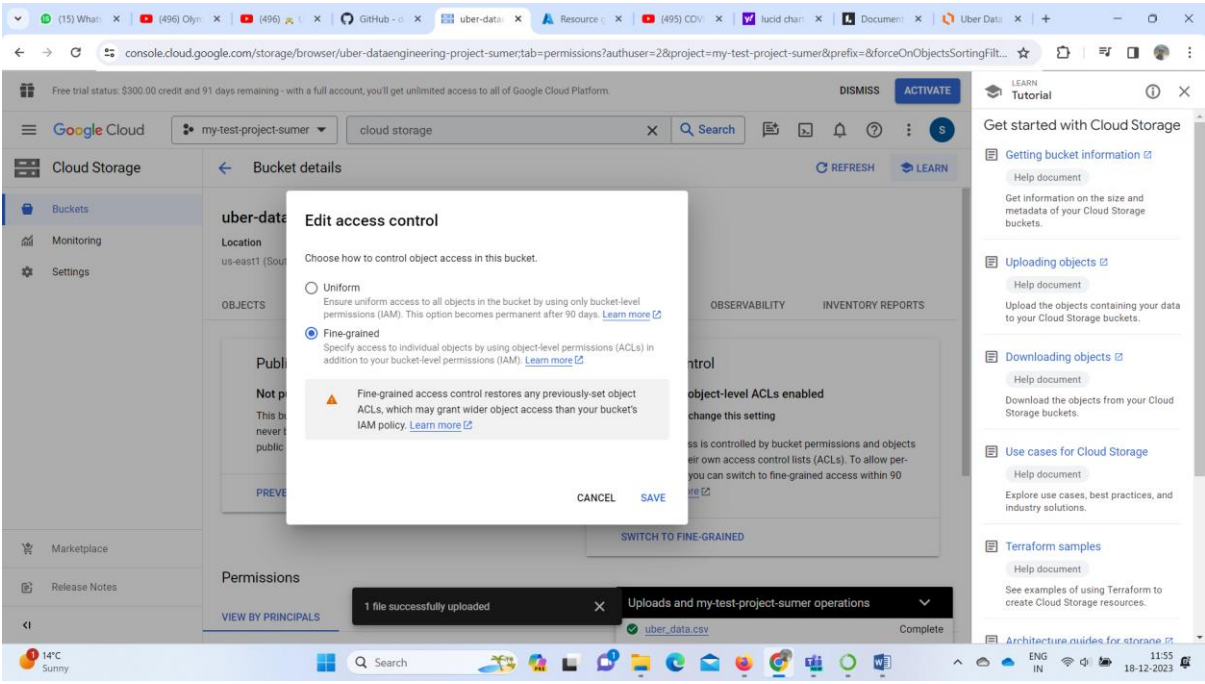


Once we are done creating the new project ,let’s create a new bucket to store our data

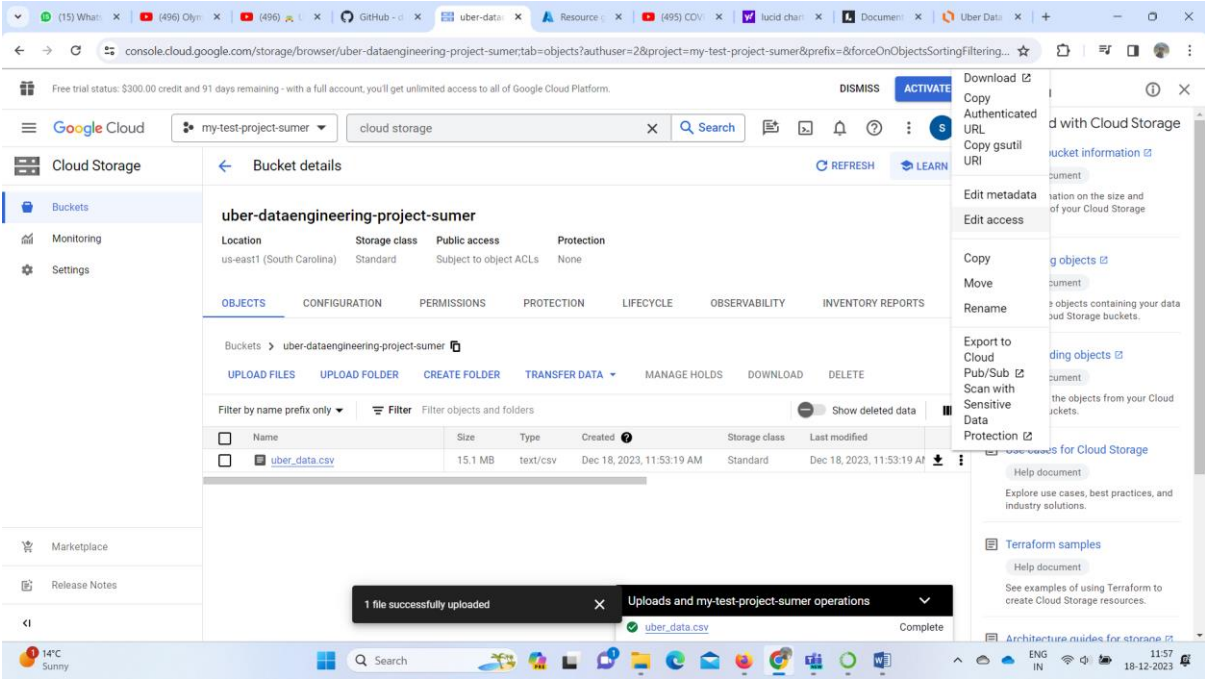


Give a name to your ,select the region .Remember to enable public access of your storage to get the URL of the data I your storage and edit the access to Fined grained

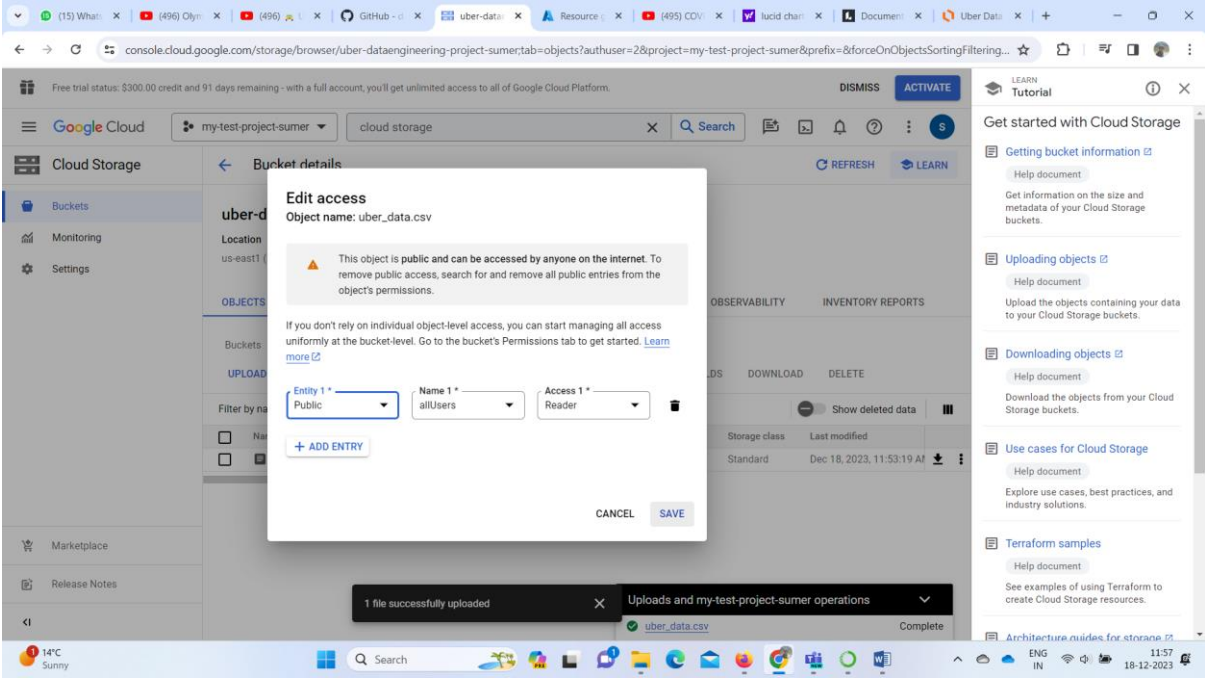




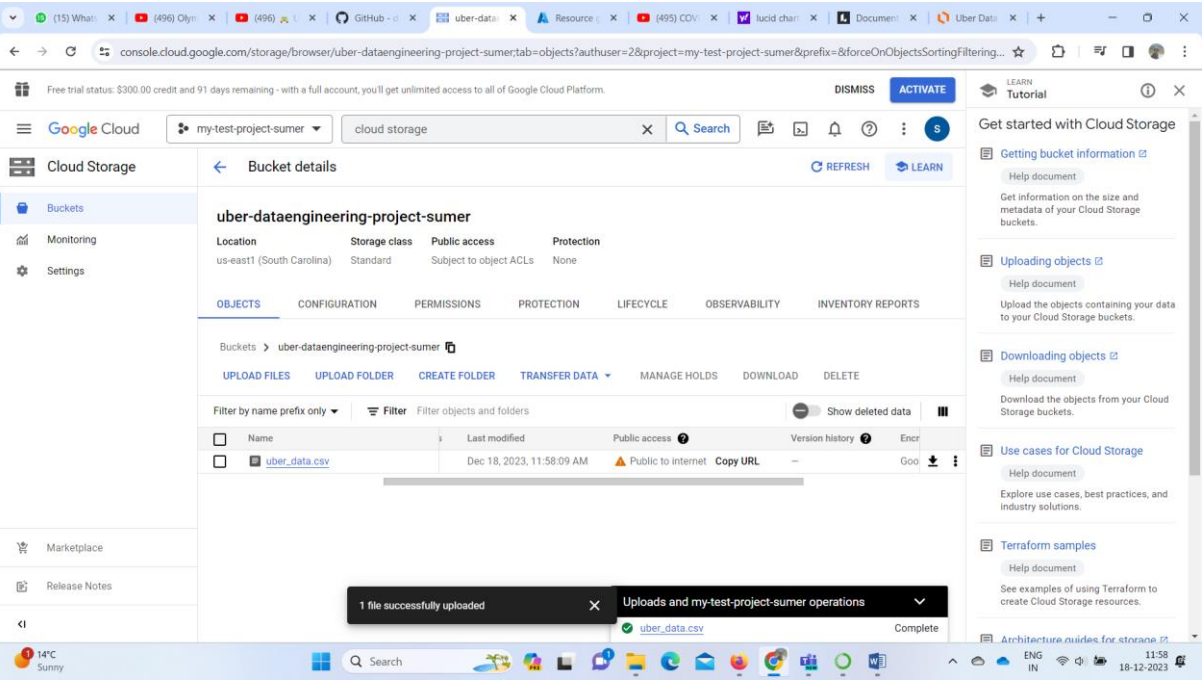
We have created the bucket



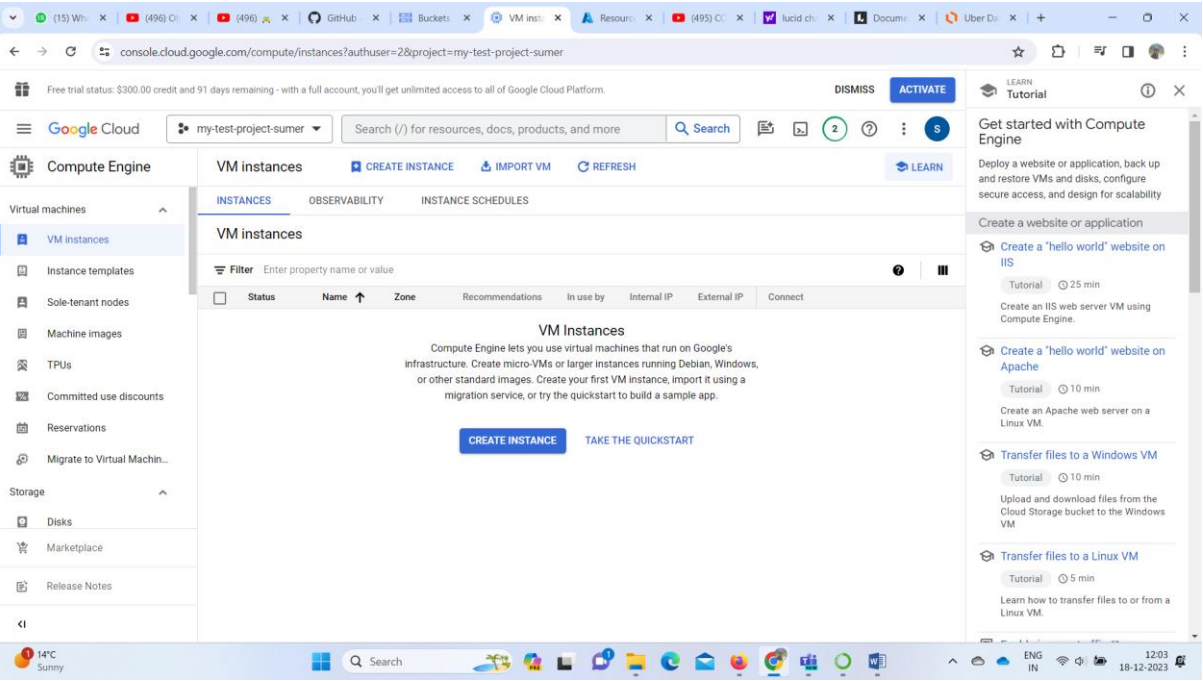
Edit the access to generate a public URL



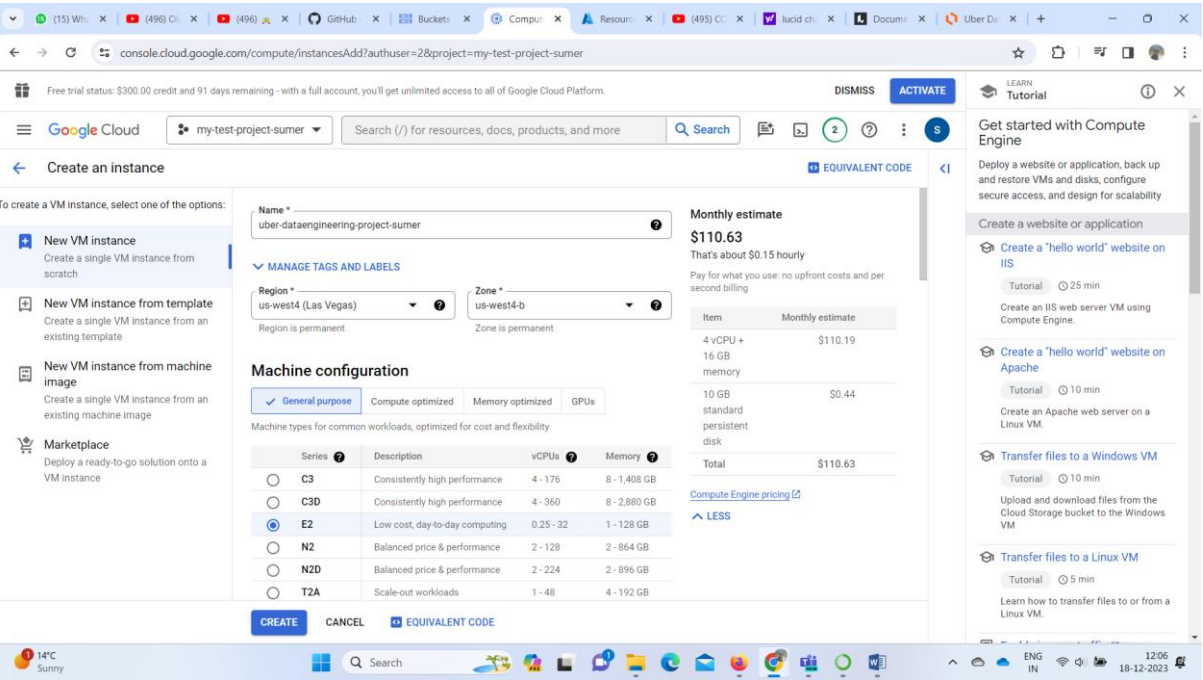
Upload the CSV in the bucket

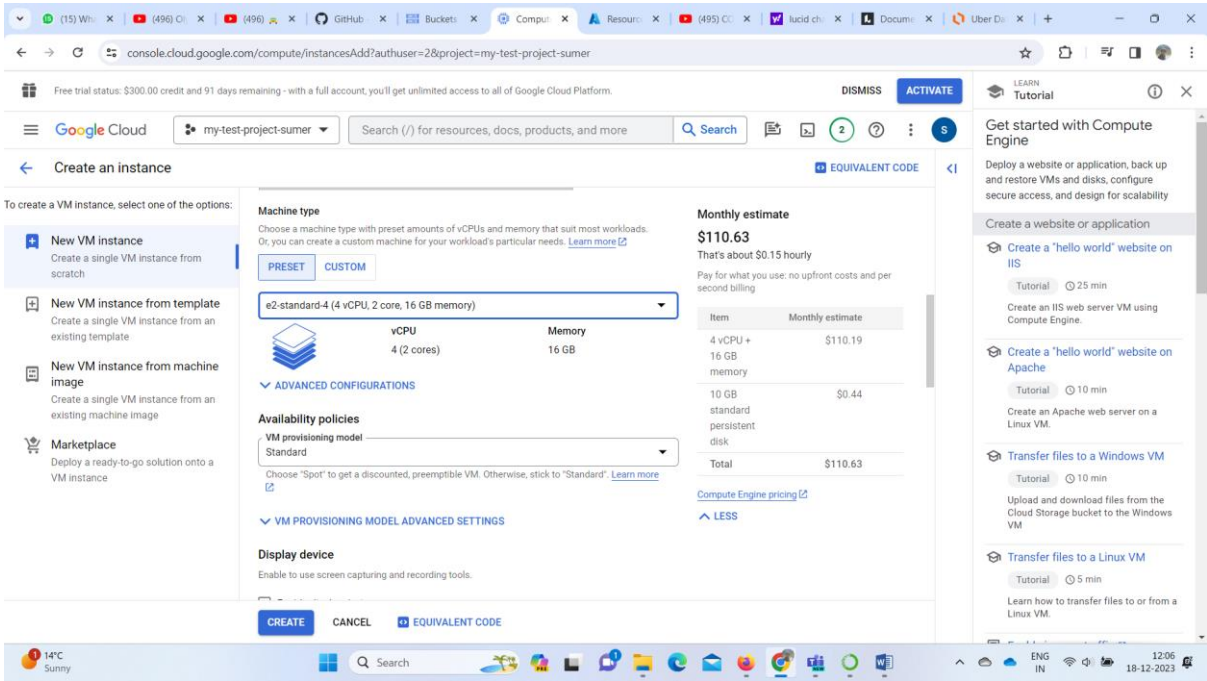


Now Let’s Start with the compute Engine .Search for google compute engine in the search box .Then create a new instance .

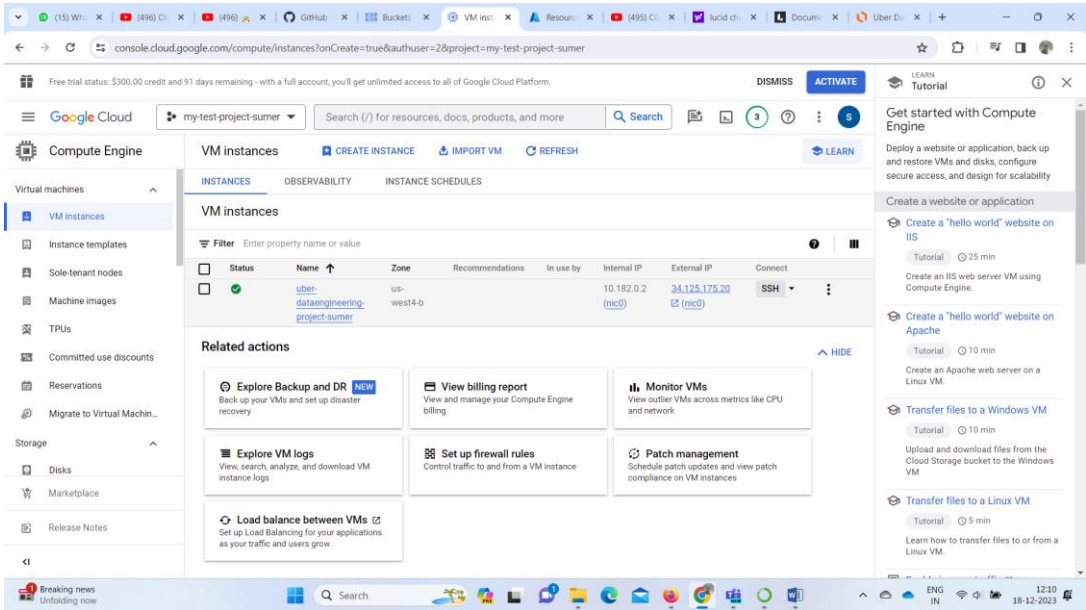
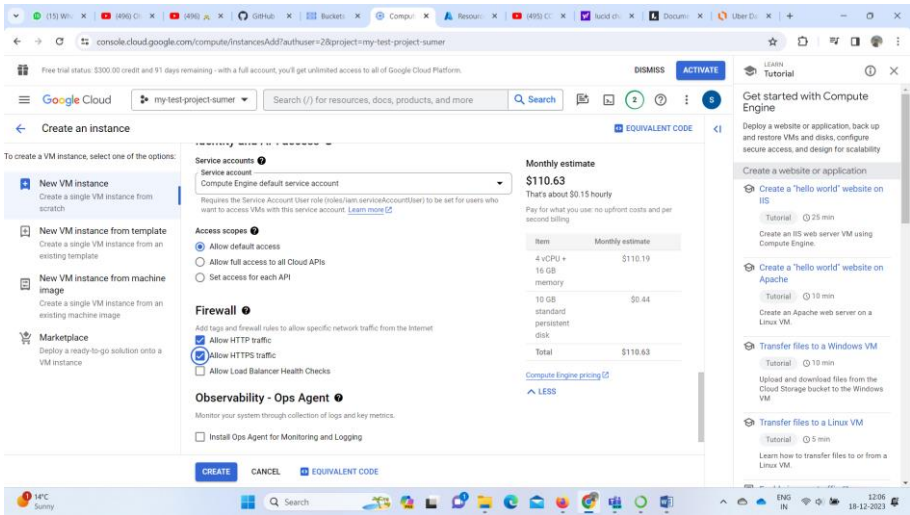


Give a name to your instance ,select the region and the machine configurations

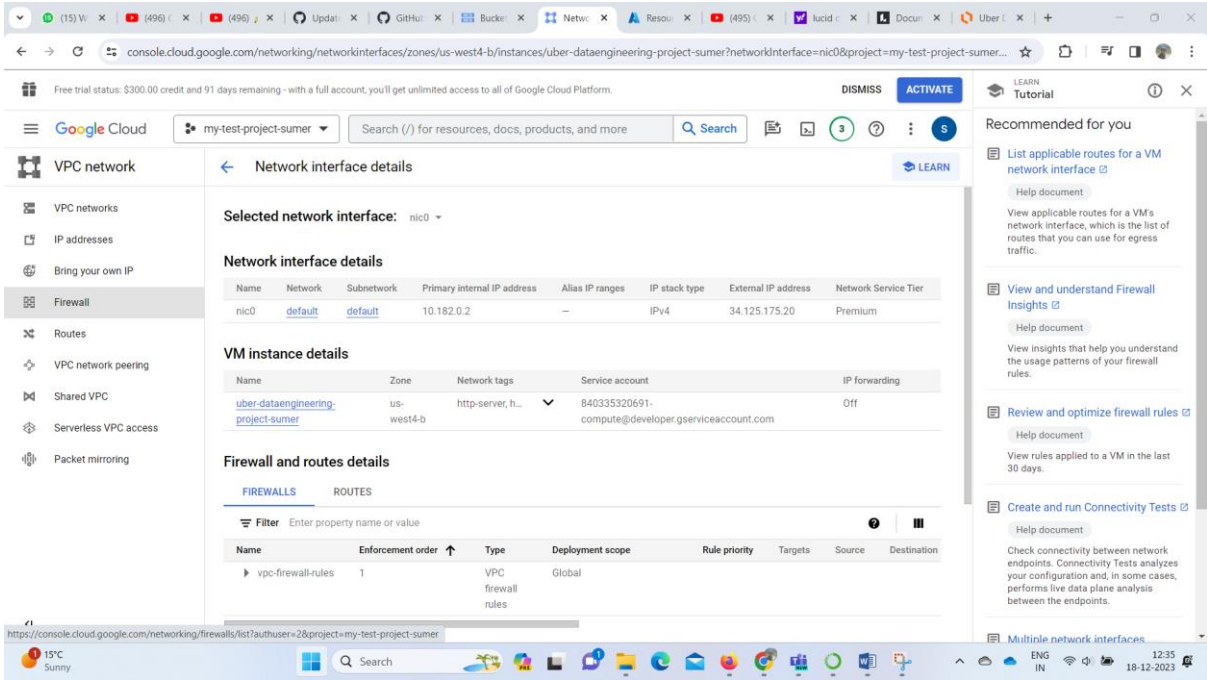




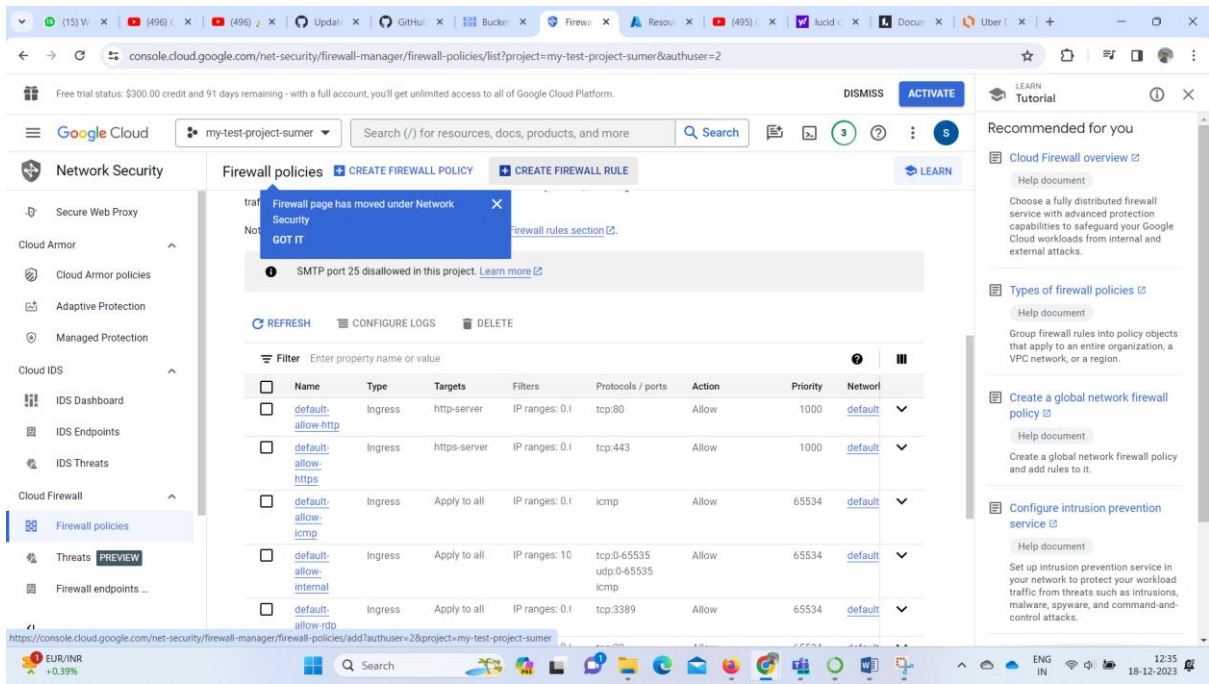
Allow the HTTP and HTTPS inbound traffic and create the machine .



Now let’s edit the Firewall rule of the instance to be able to connect to mage ai .to to the firewall pane ,click on ‘nic0’.



And allow TCP port 6789(mageai port) under the inbound rules



now let’s work on our newly constructed instance ,by clicking on SSH
 Install the important packages and mage on our new instance .Below is the code for that

```
# Install Python and pip
```

```
sudo apt-get install update
```

```
sudo apt-get update
```

```
sudo apt-get install python3-distutils
```

```
sudo apt-get install python3-apt
```

```
sudo apt-get install wget
```

```
wget https://bootstrap.pypa.io/get-pip.py
```

```
sudo python3 get-pip.py
```

```
# Install Mage
```

```
sudo pip3 install mage-ai
```

```
# Install Pandas
```

```
sudo pip3 install pandas
```

```
# Install Google Cloud Library
```

```
sudo pip3 install google-cloud
```

```
sudo pip3 install google-cloud-bigquery
```

Once mage is pip installed we have to start mage with a name of the project

```

228.7/228.7 KB 22.2 MB/s eta 0:00:00
Downloading cachetools-5.3.2-py3-none-any.whl (9.3 kB)
Installing collected packages: rsa, pyasn1-modules, googleapis-common-protos, google-crc32c, cachetools, google-resumable-media, google-auth, google-api-core, google-cloud-core, google-cloud-bigquery
Successfully installed cachetools-5.3.2 google-api-core-2.15.0 google-auth-2.25.2 google-cloud-bigquery-3.14.1 google-cloud-core-2.4.1 google-crc32c-1.5.0 google-resumable-media-2.7.0 googleapis-common-protos-1.62.0 pyasn1-modules-0.3.0 rsa-4.9
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
sumerrajkumarpariani2000@uber-dataengineering-project-sumer:~$ mage start uberdataprotect

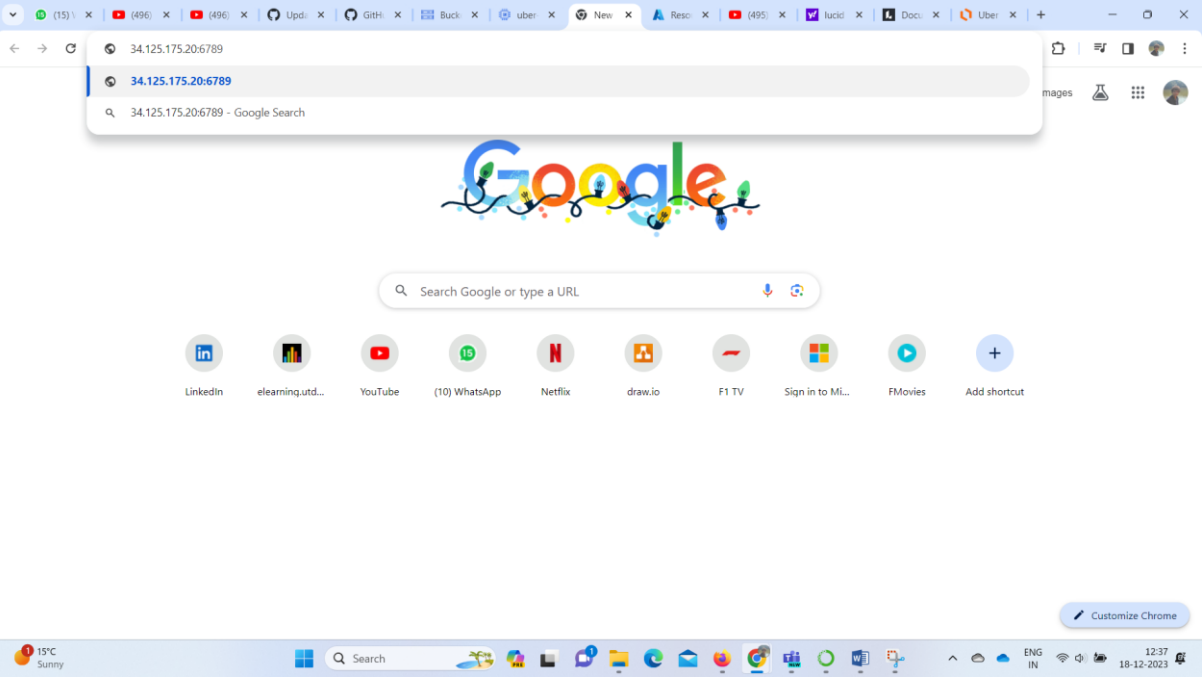
```

This will give you a port number on which the mage instance is working

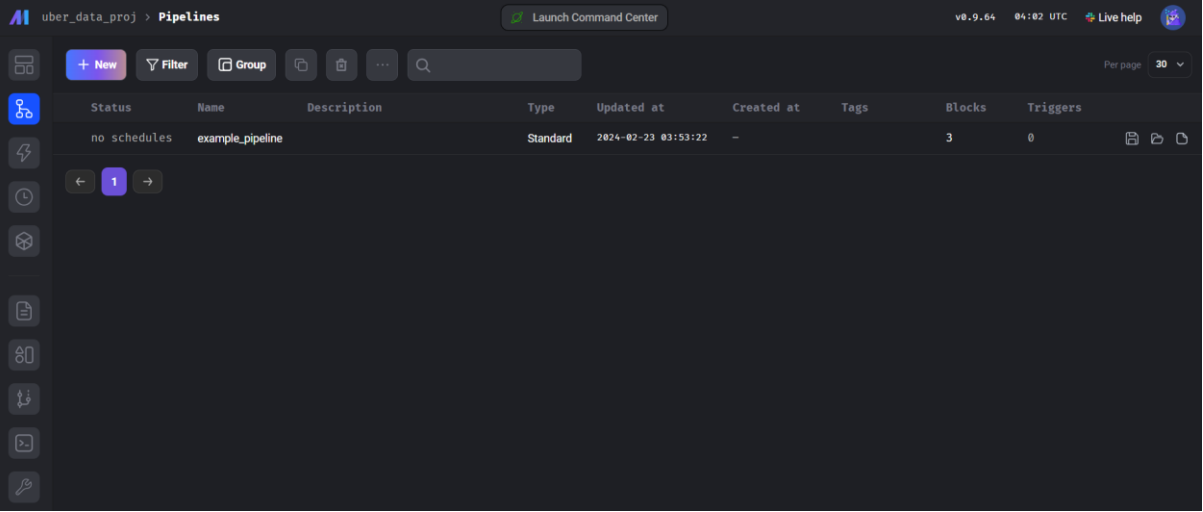
port 6789

```
ation tables
Checking port 6789...
INFO [alembic.runtime.migration] Running upgrade 643b6e65e814 -> 1f9353eddbc6, Add secrets table
INFO:mage_ai.server.server:Mage is running at http://localhost:6789 and serving project /home/sumerrajkumarpari
```

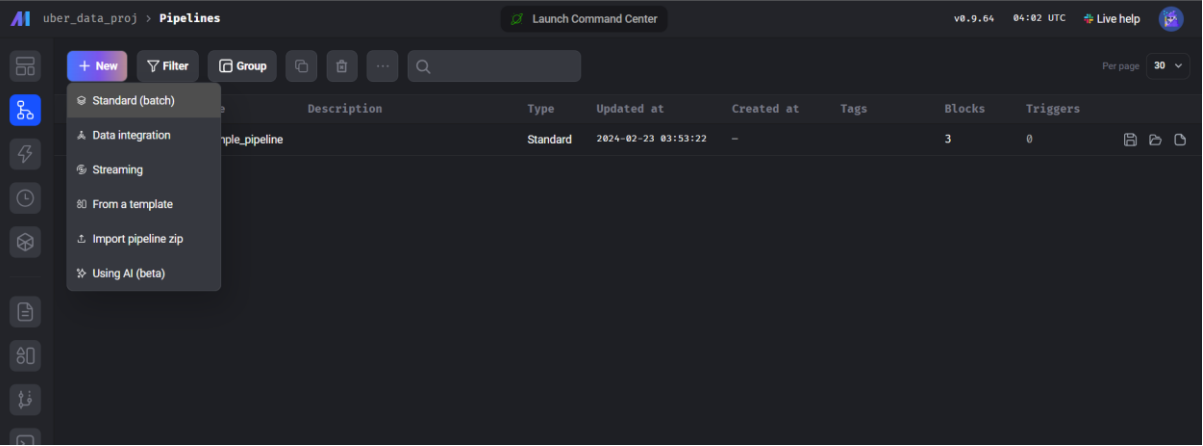
once we have our mage running let’s connect to mage UI .Copy the external IP of the instance and connect to the 6789 TCP port



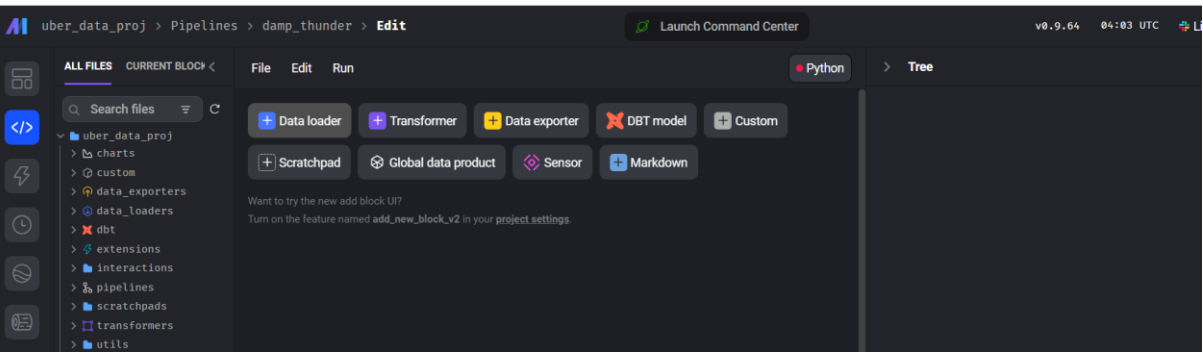
Here we are on our Mage UI



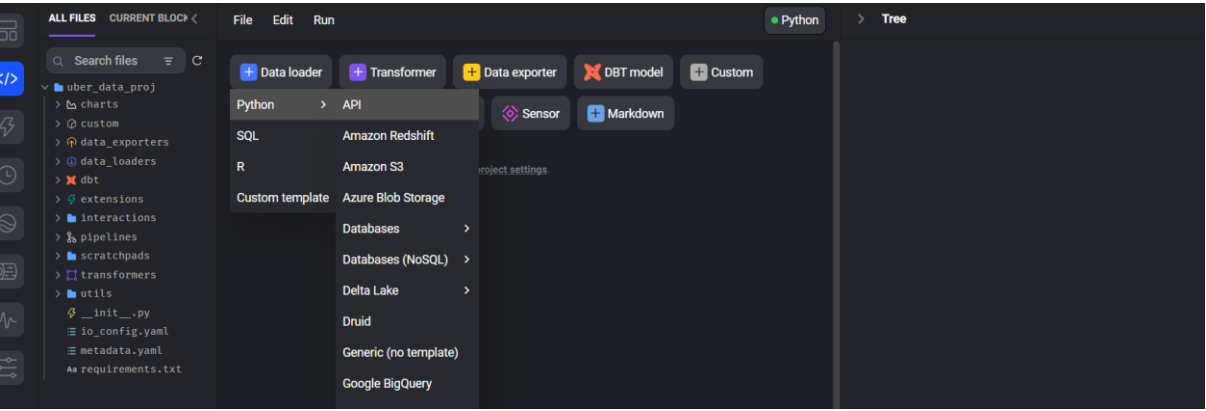
Start by creating a new standard (Batch Process)



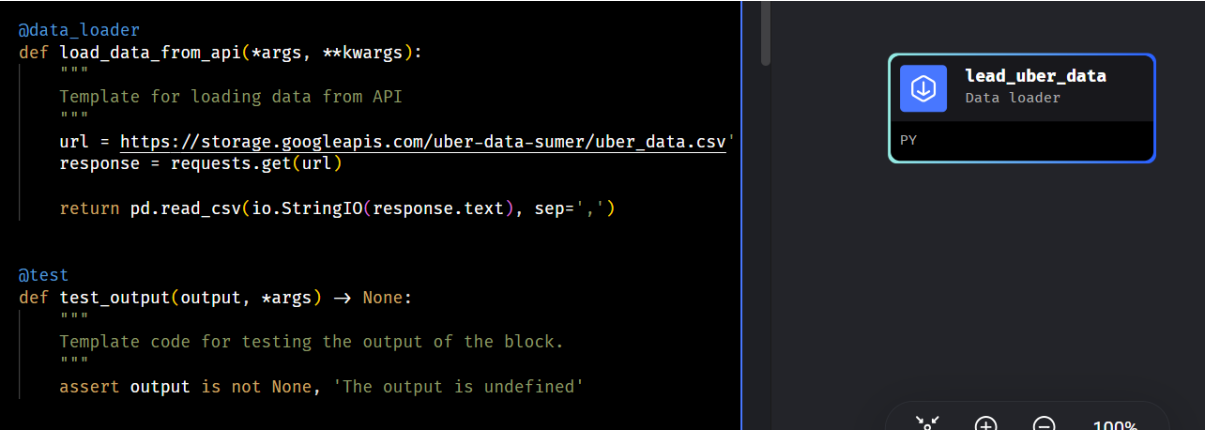
to load the Data click on the Data loader option



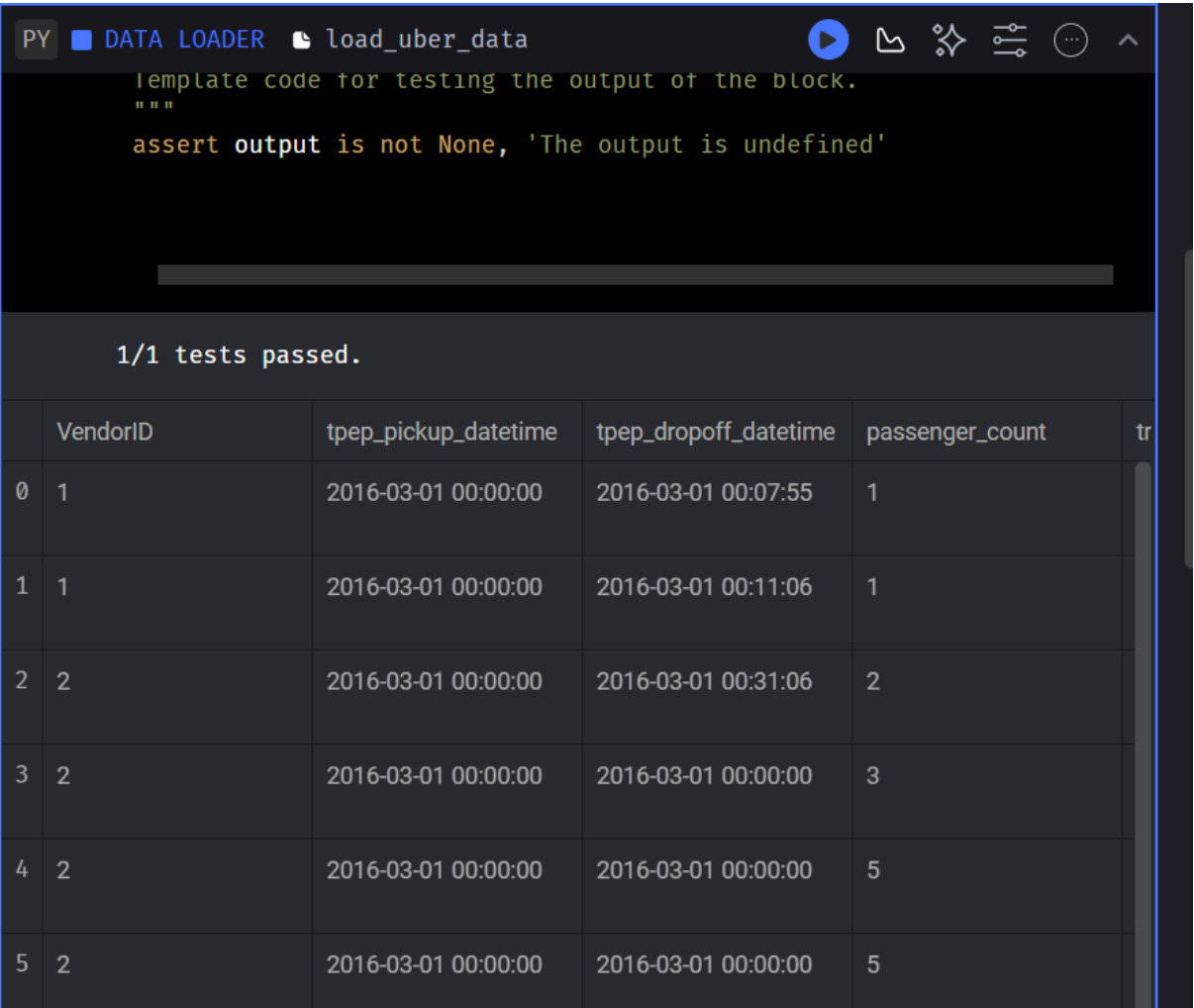
Select Python a language and then through API because we have the data in your google cloud storage .



copy the URL of the data in google cloud storage and paste it under data loader code of Mage UI

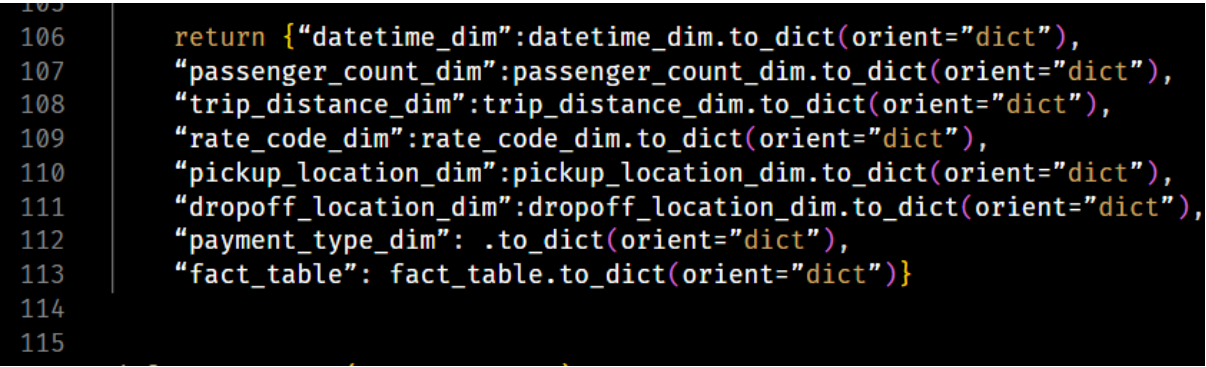
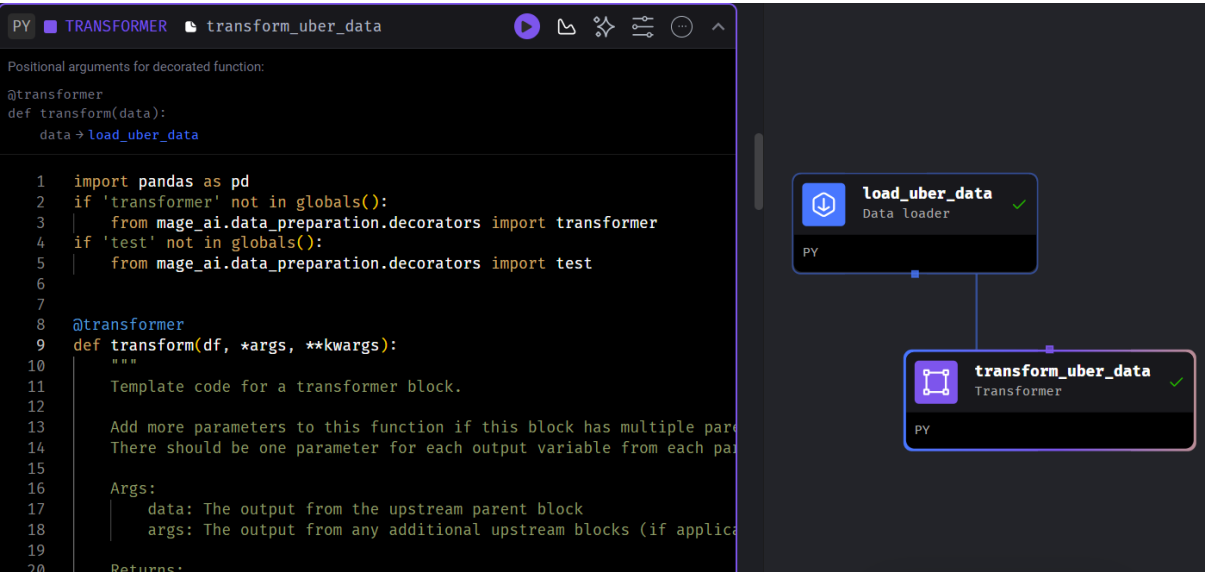


Run the code and access the data

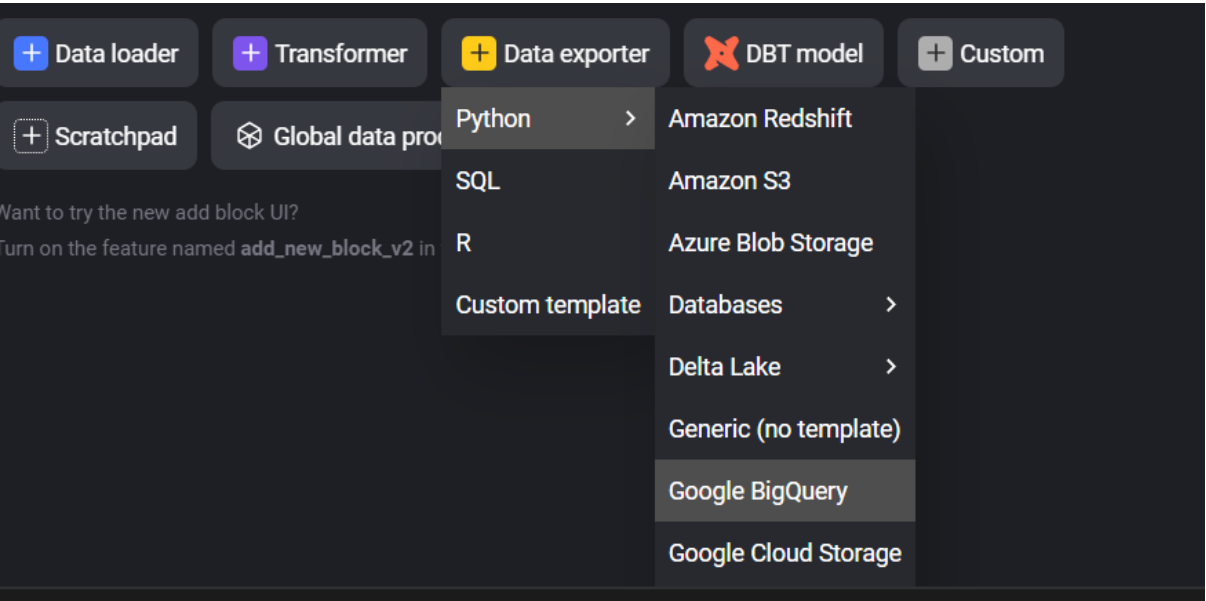


Now we move to the transformation park .Click on the transform option to open the mage code for transformation .Apply the transformation code in the section given there .Remember to return all the tables at the end .

To have provide my transformation code visit this Link -https://github.com/SumerPariani/GCP/blob/main/uber_data_analysis.ipynb

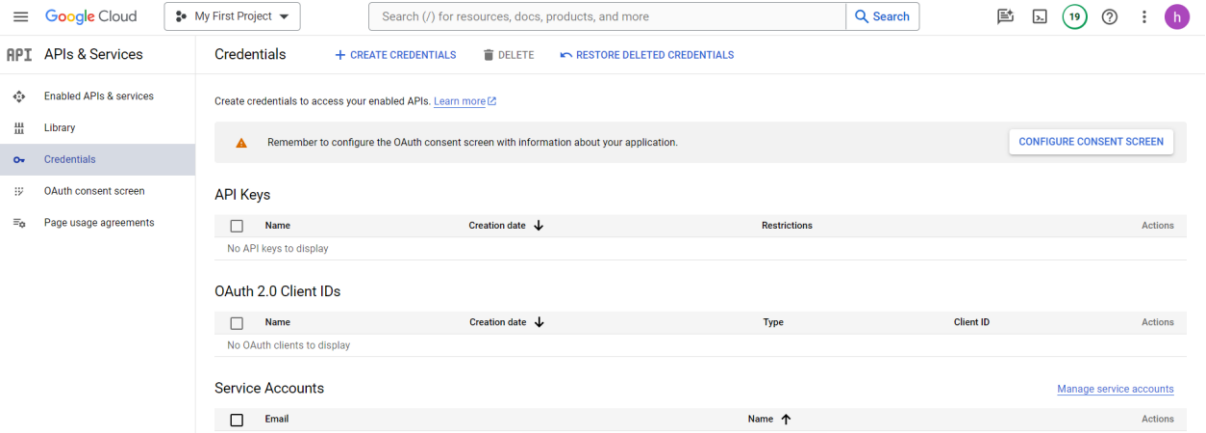


Now let’s move exporter part .We need to export the data to Google BigQuery to be able to perform querying on this data



we need to provide credentials to the YAML.io file in MAGE UI to be able to connect with a Big query to generate the access credentials on google cloud console .

Go to API’s and services and the credentials pane .Click on create Credentials for service account option



```
PY ■ DATA EXPORTER export_uber_data ← 1 parent

Positional arguments for decorated function:
@data_exporter
def export_data(data):
    data → transform_uber_data

from mage_ai.settings.repo import get_repo_path
from mage_ai.io.bigquery import BigQuery
from mage_ai.io.config import ConfigFileLoader
from pandas import DataFrame
from os import path

if 'data_exporter' not in globals():
    from mage_ai.data_preparation.decorators import data_exporter

@data_exporter
def export_data_to_big_query(data: DataFrame, **kwargs) → None:
    """
    Template for exporting data to a BigQuery warehouse.
    Specify your configuration settings in 'io_config.yaml'.

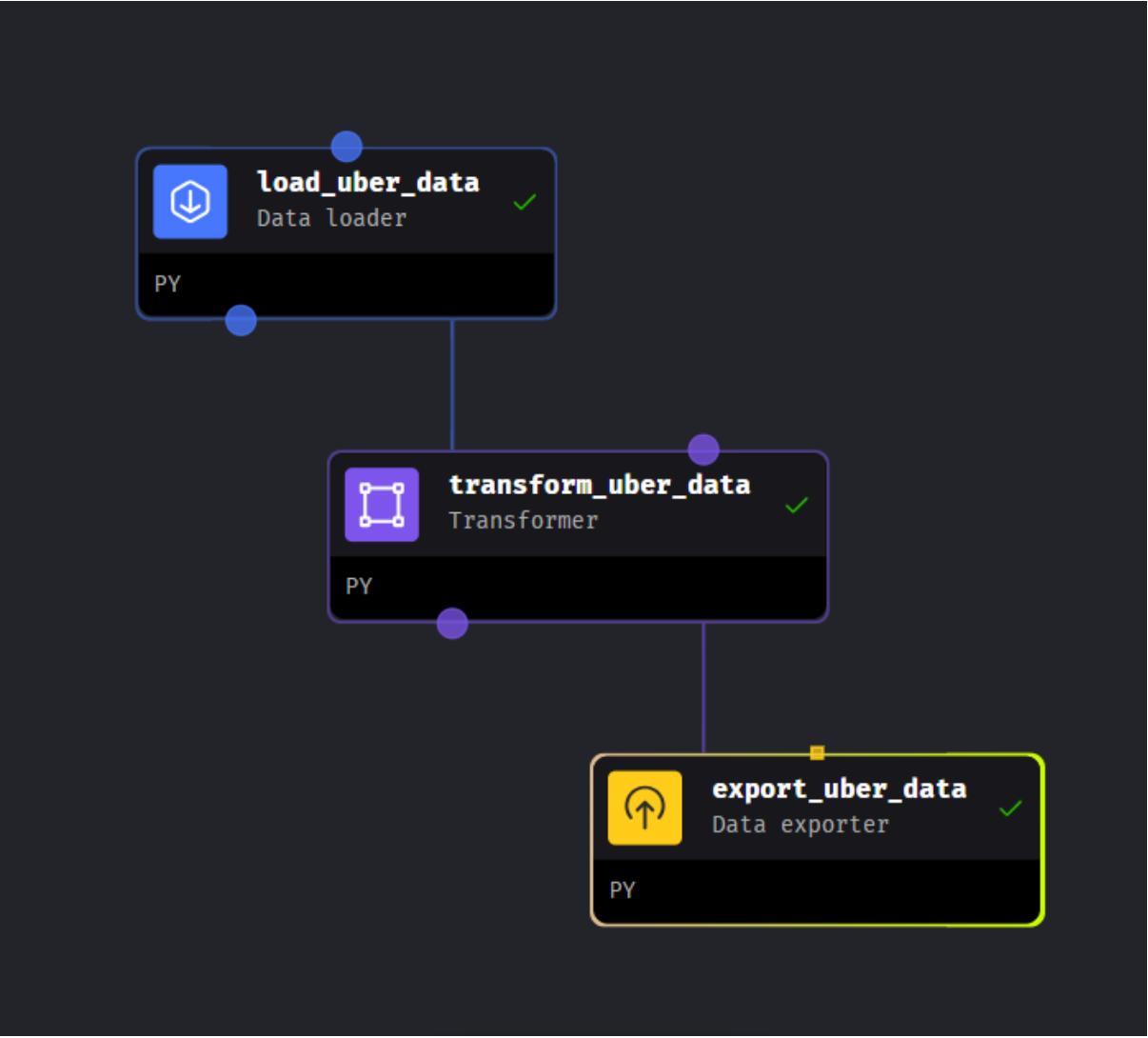
    Docs: https://docs.mage.ai/design/data-loading#bigquery
    """
    config_path = path.join(get_repo_path(), 'io_config.yaml')
    config_profile = 'default'

    for key,value in data.items():
        table_id = 'corded-bivouac-415001.uber_data.{}'.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
```

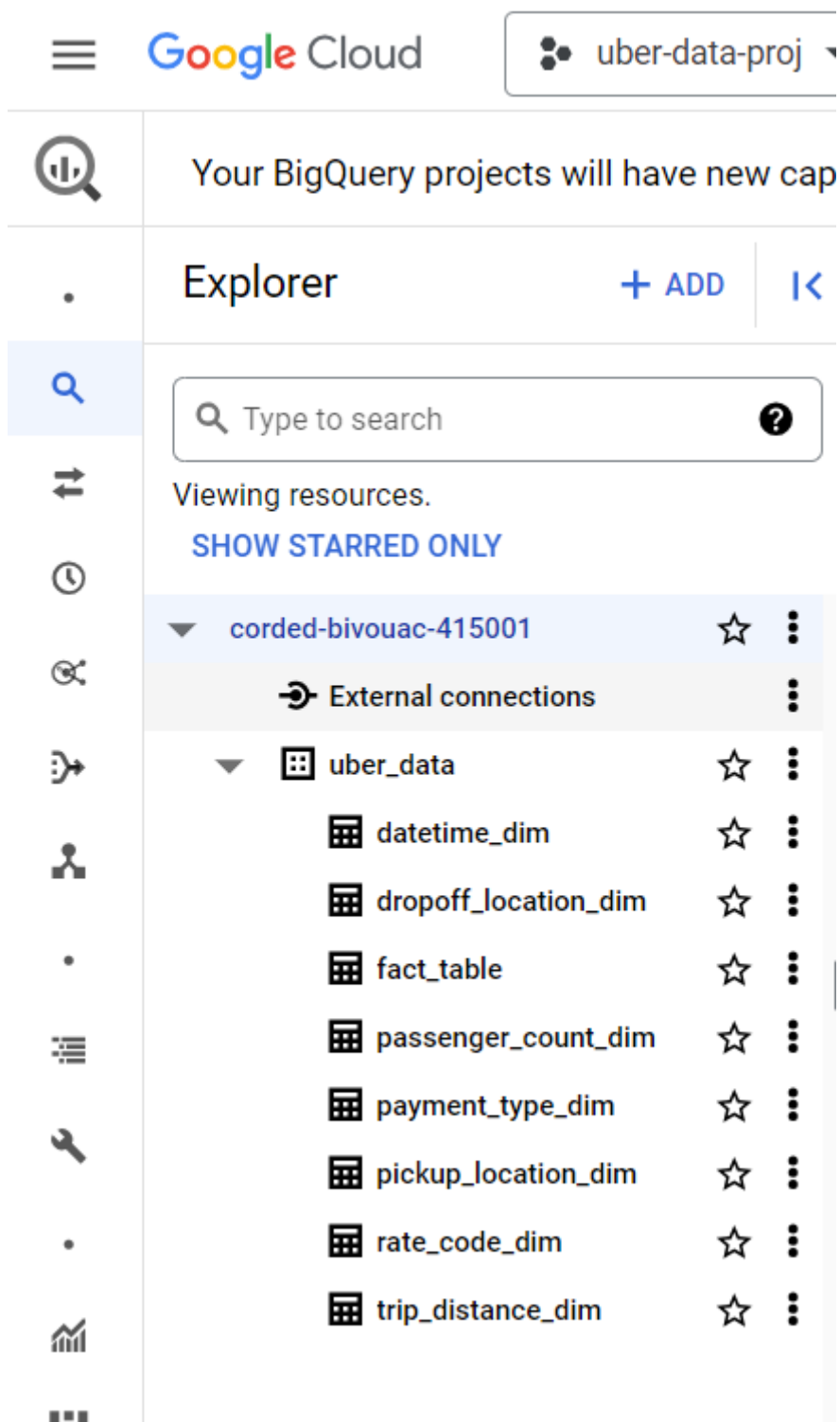
And Woohoo we have successfully exported the data to our BigQuery page.

```
PY ■ DATA EXPORTER export_uber_data ← 1 parent

BigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.datetime_dim' ...
DONEBigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.passenger_count_dim' ...
DONEBigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.trip_distance_dim' ...
DONEBigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.rate_code_dim' ...
DONEBigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.pickup_location_dim' ...
DONEBigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.dropoff_location_dim' ...
DONEBigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.payment_type_dim' ...
DONEBigQuery initialized
└ Connecting to BigQuery warehouse ... DONE
└ Exporting data to table 'corded-bivouac-415001.uber_data.fact_table' ...
DONE
```



Open the BigQuery Services on the Google Console and we can see the tables are exported to the database in BigQuery



we can run queries and get insights on the Data in Big Query. We can also create relationships between tables and create new tables in our database on BigQuery .

Below are some of the queries which I ran to get insights on the Uber data Analysis project.

Finally we have to make a Power BI dashboard to Visualize the Data we analysed on BigQuery .

Here is the PowerBi Dashboard I created .

