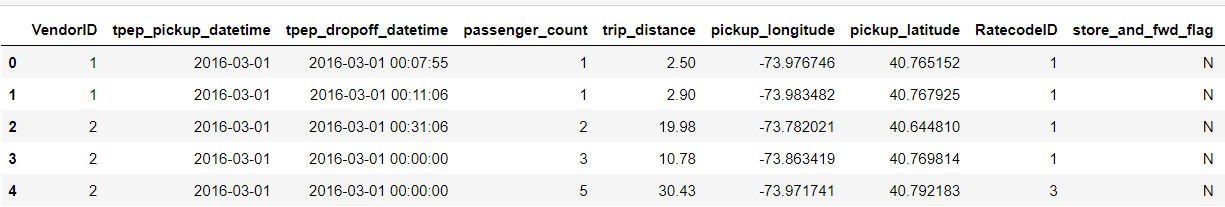
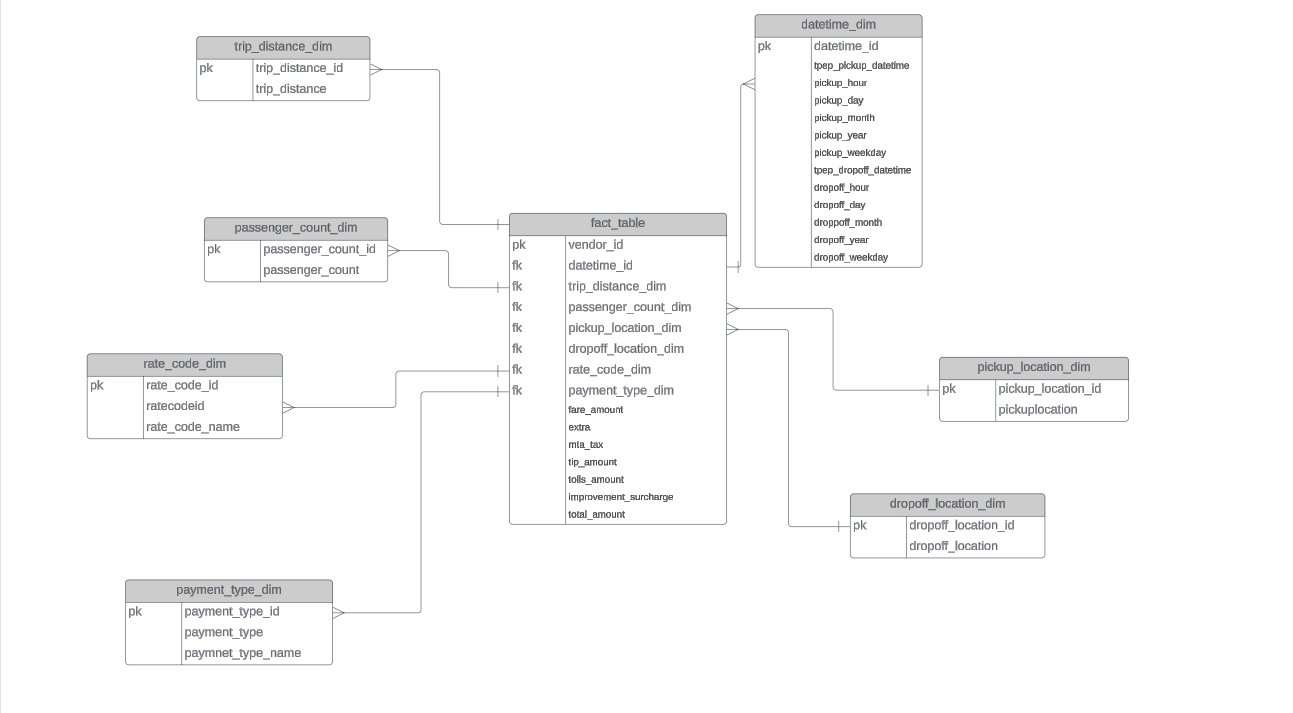
  
  
Here I present the end-to-end Google Data Pipeline which I created to analyse the Uber data and visualize it on a Power BI Dashboard.  
  
The resources and Tools I used are -  
Raw data –Acquired the Raw data through open source portals  
Draw.io-Used Draw.io to perform Dimensional modelling of the 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   


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

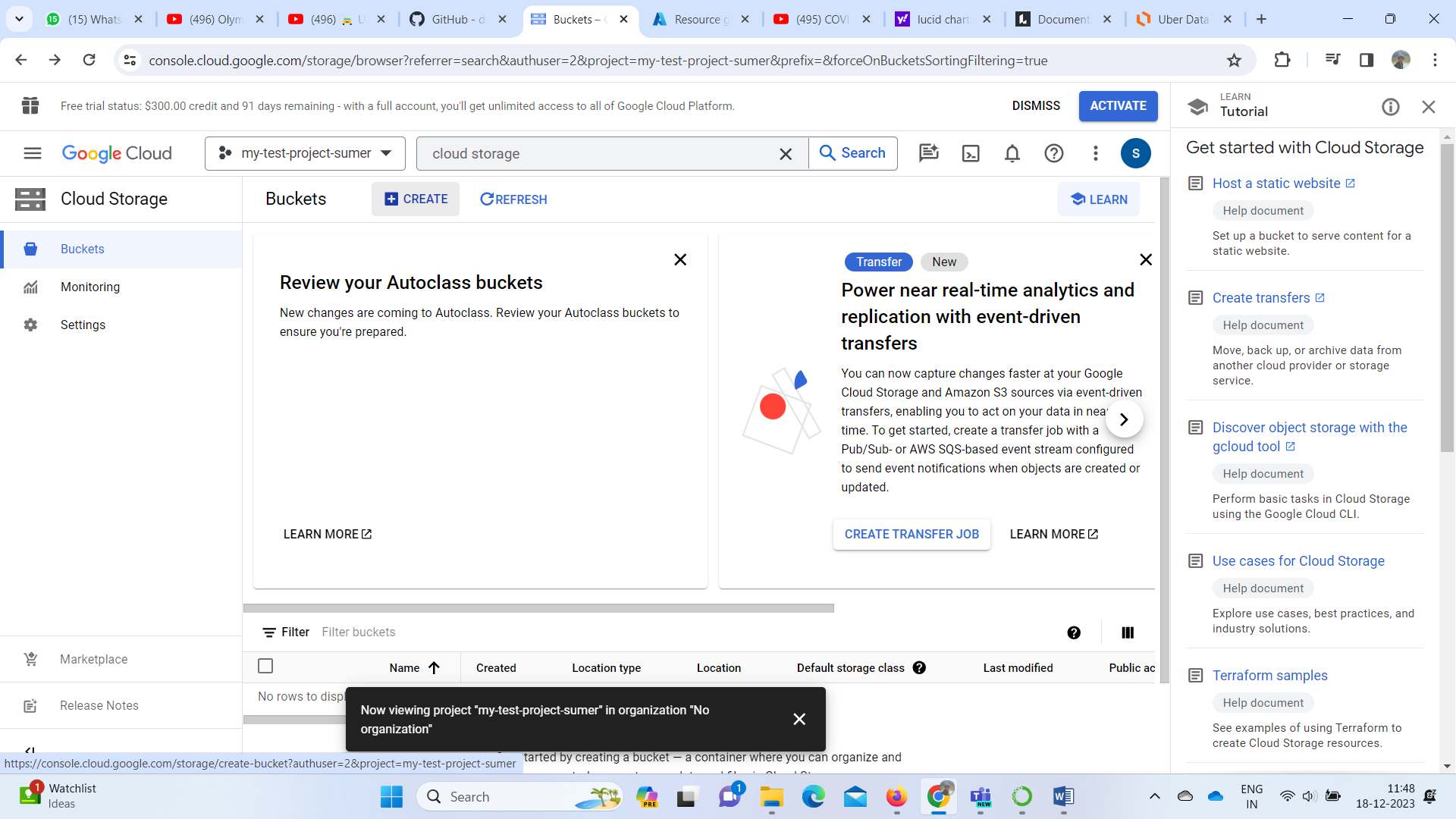
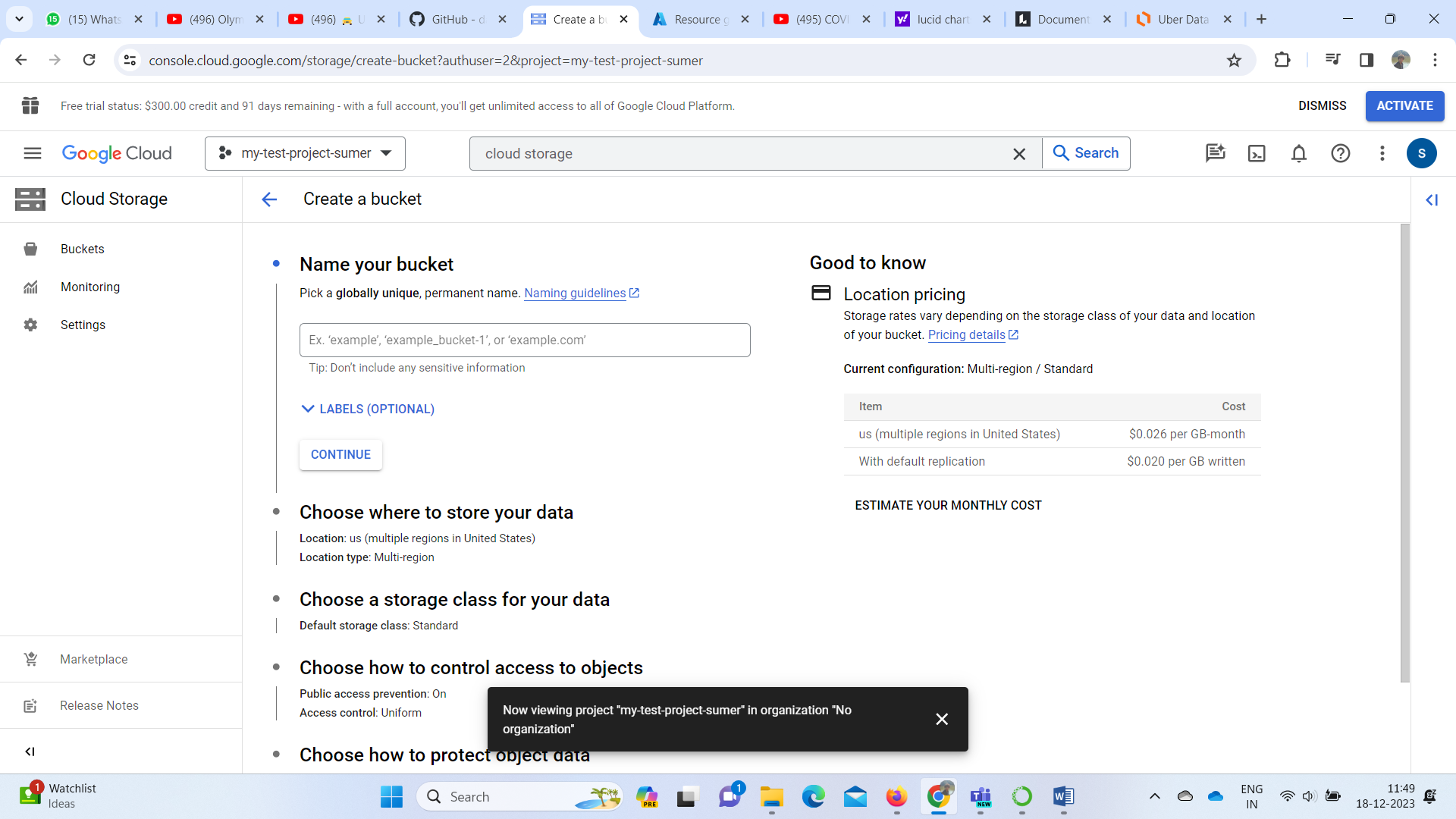
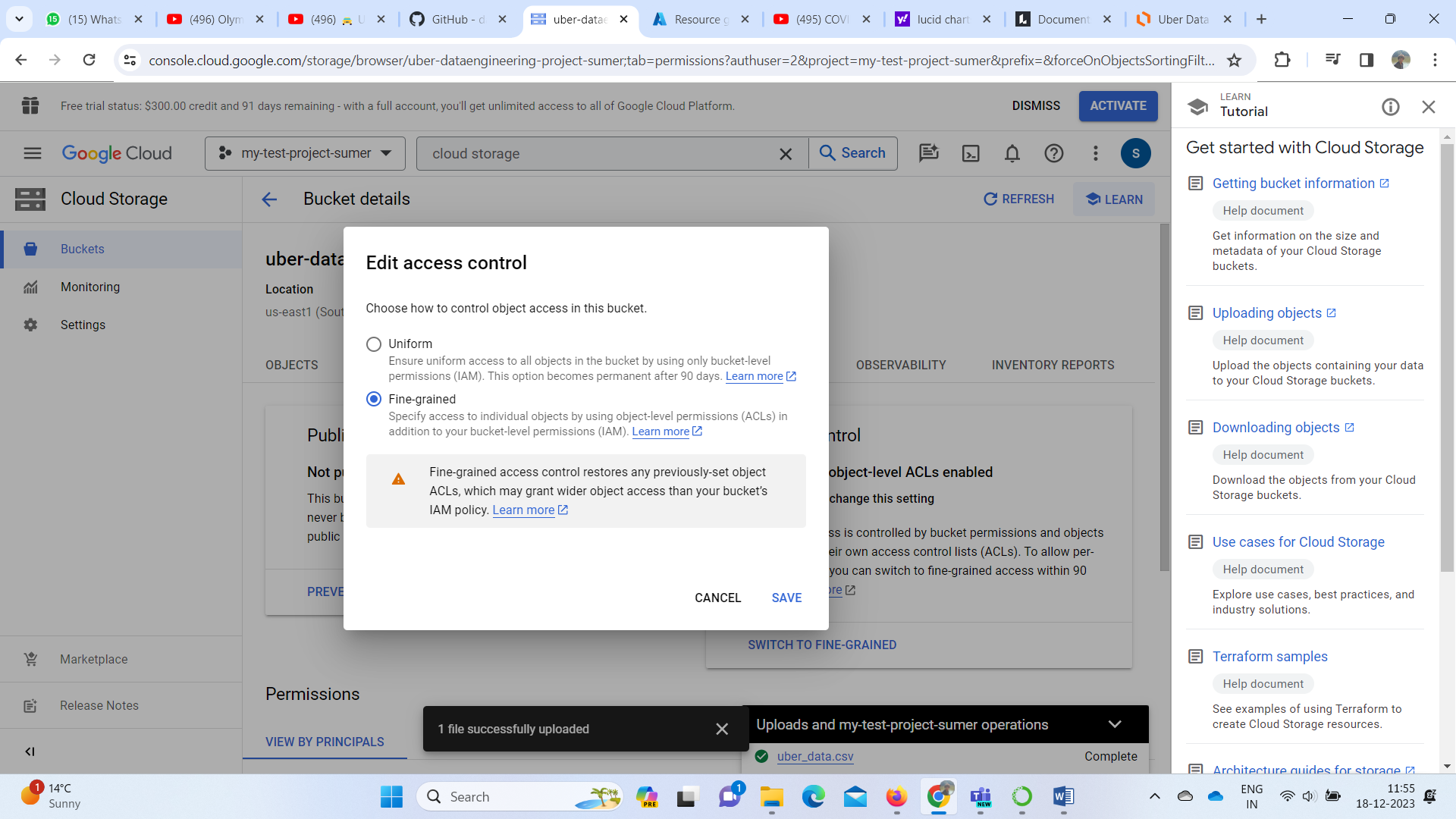
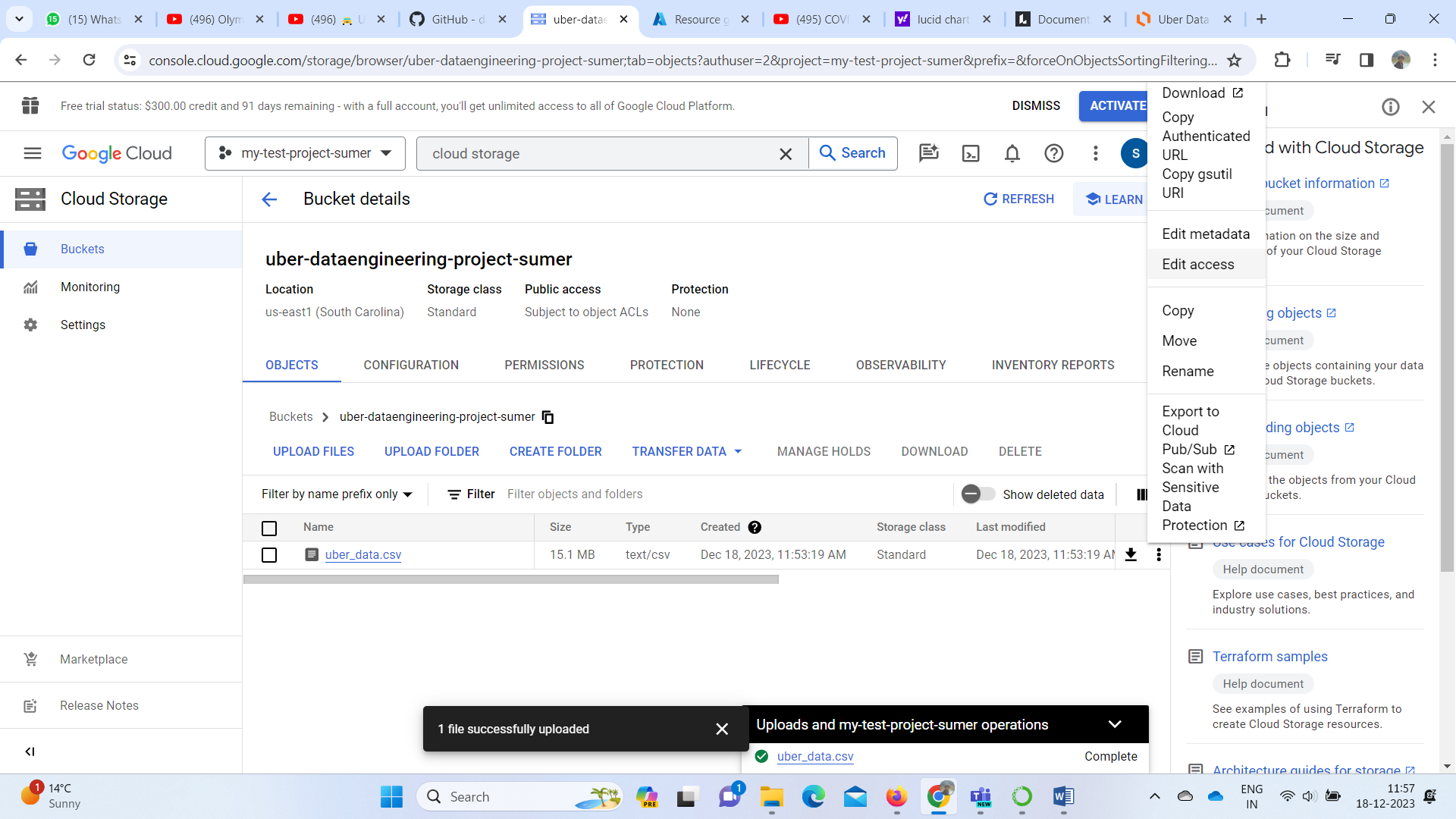
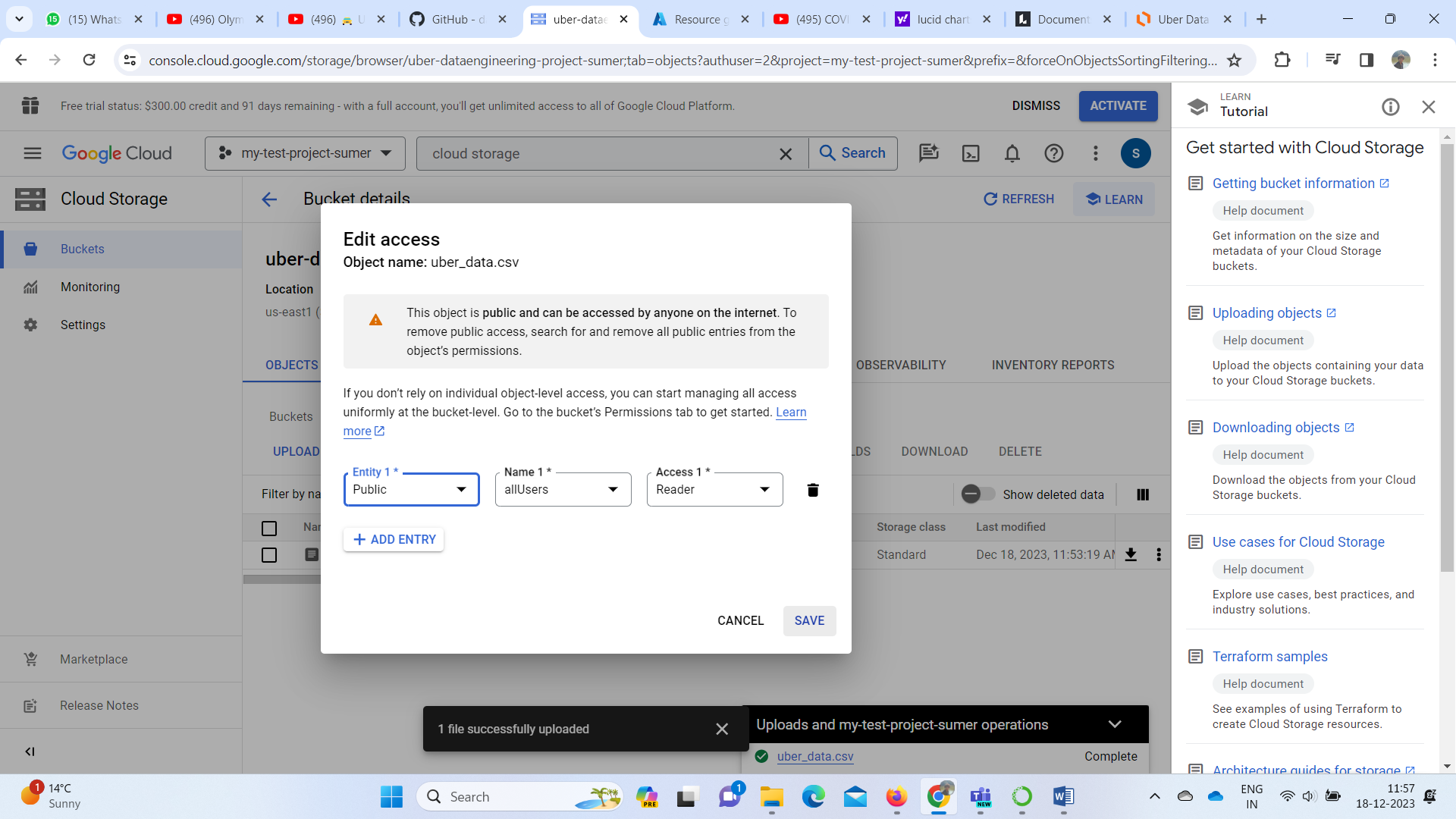
3. dropoff location table

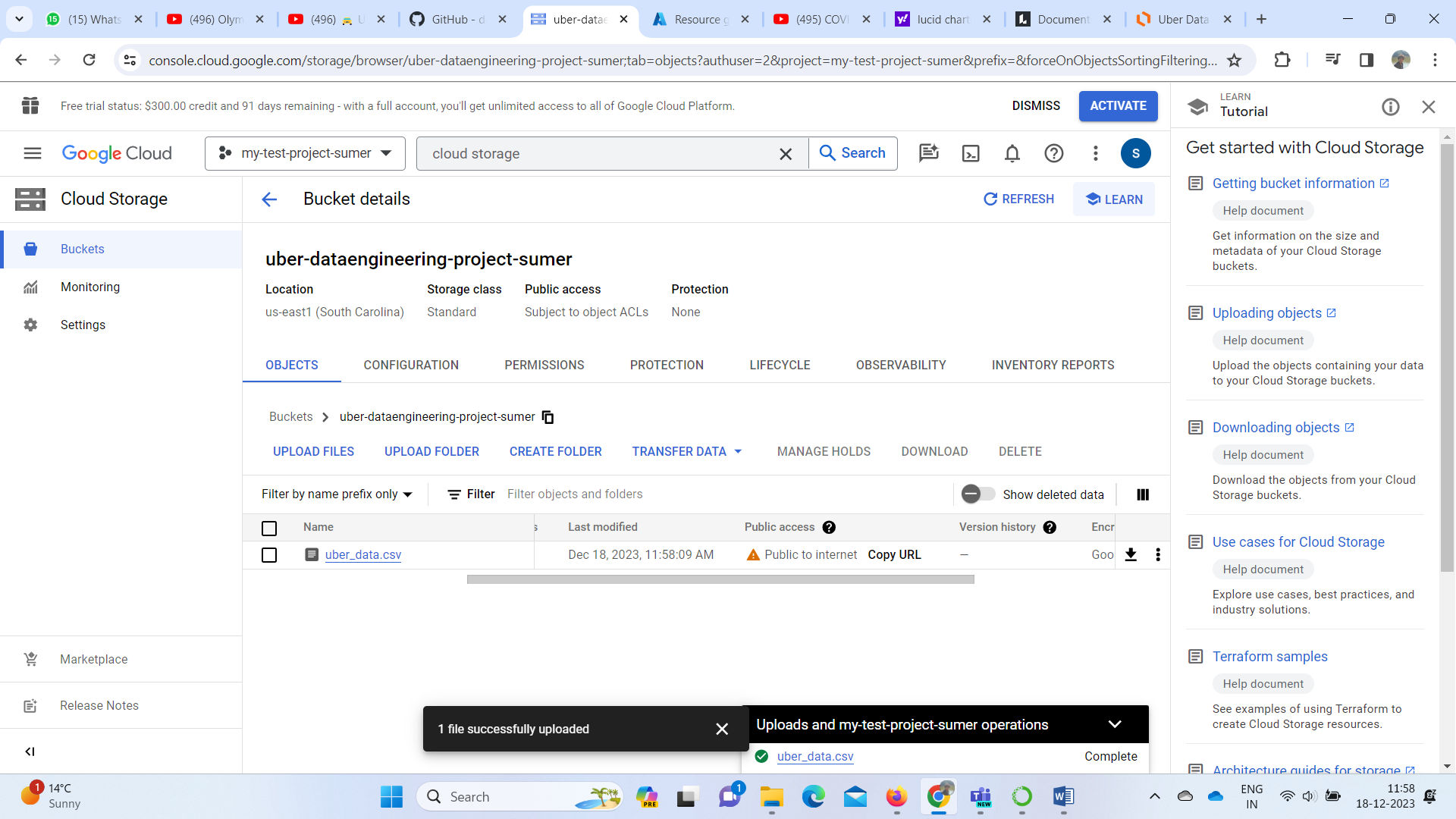
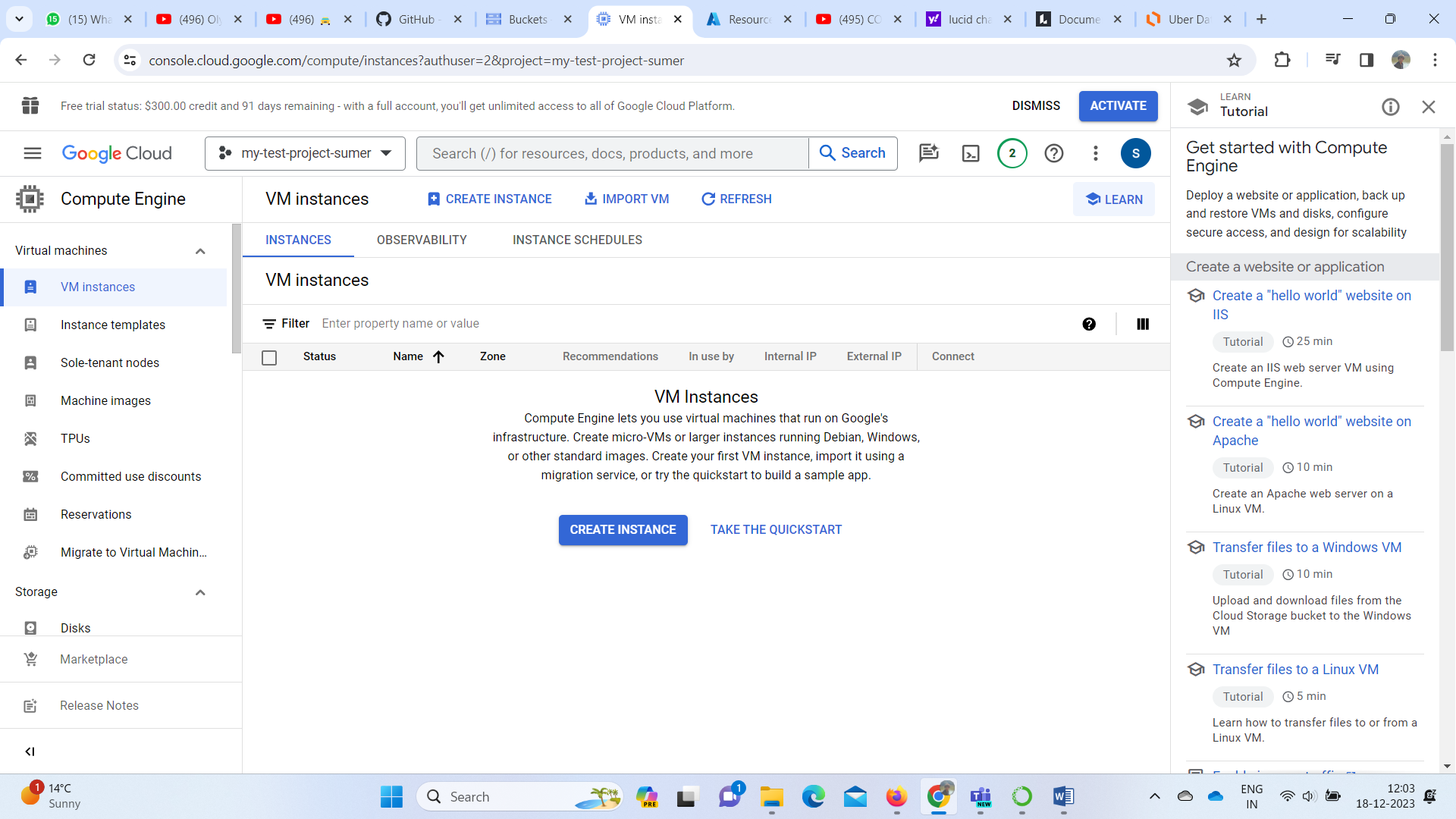
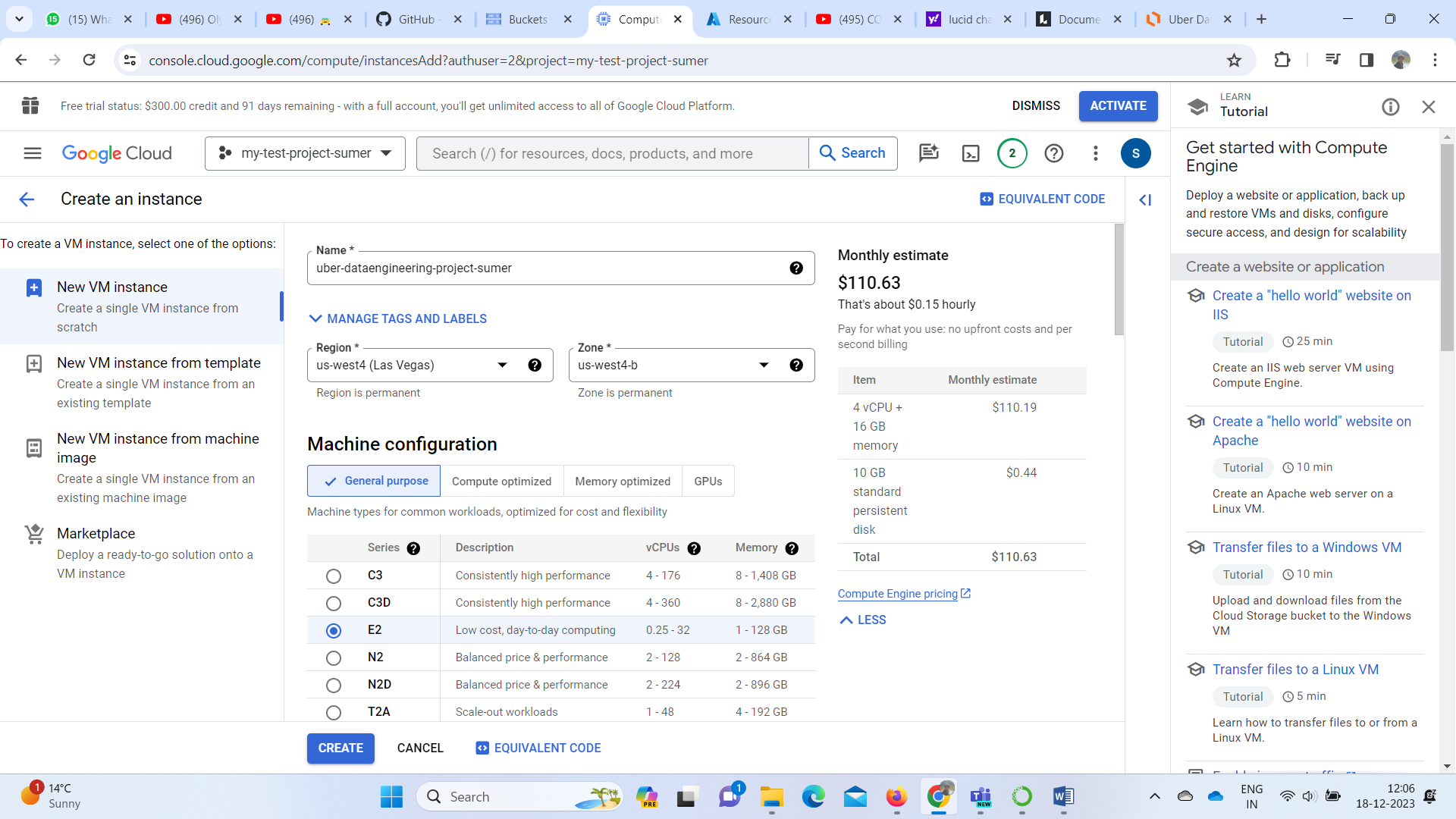
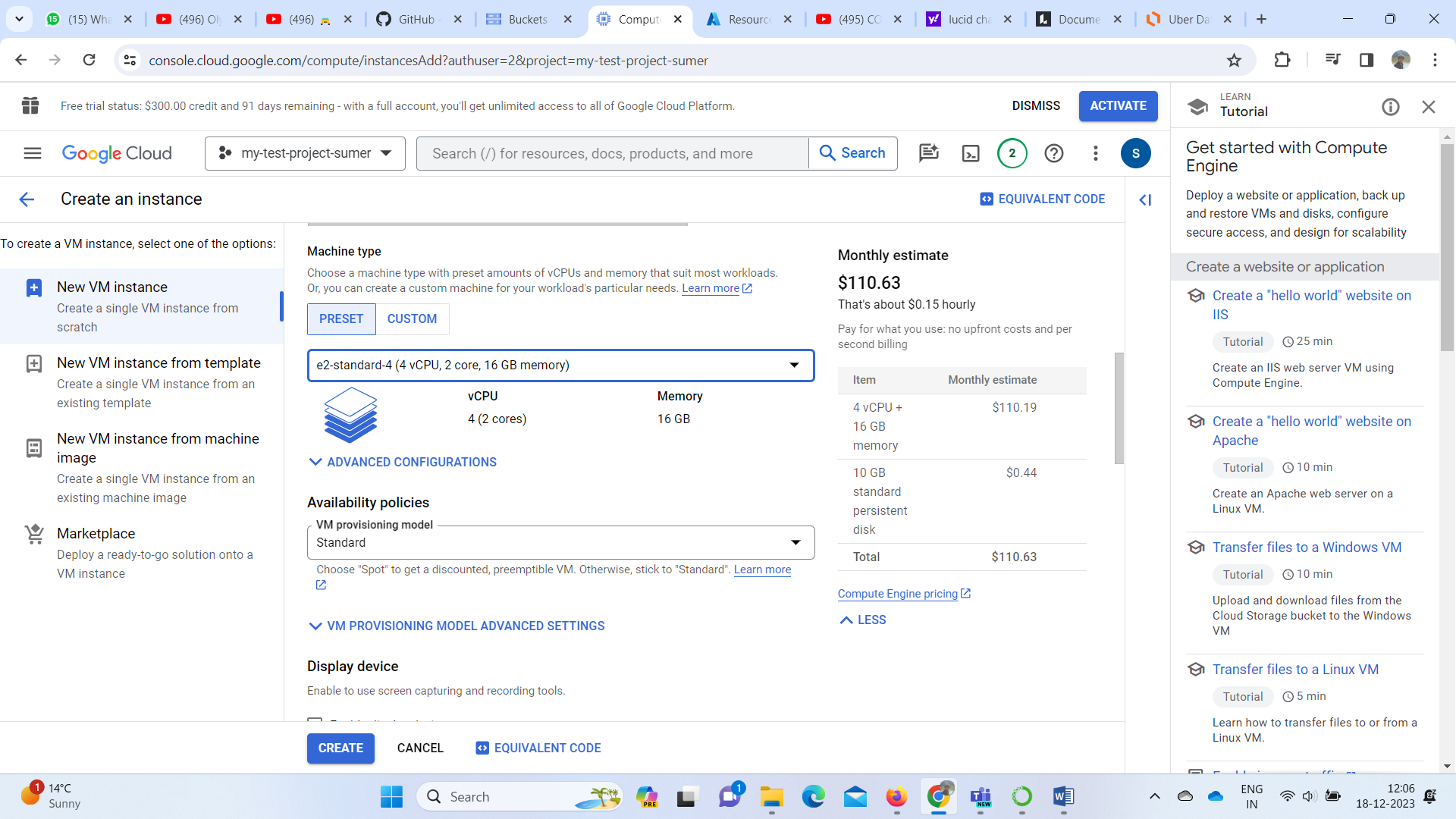
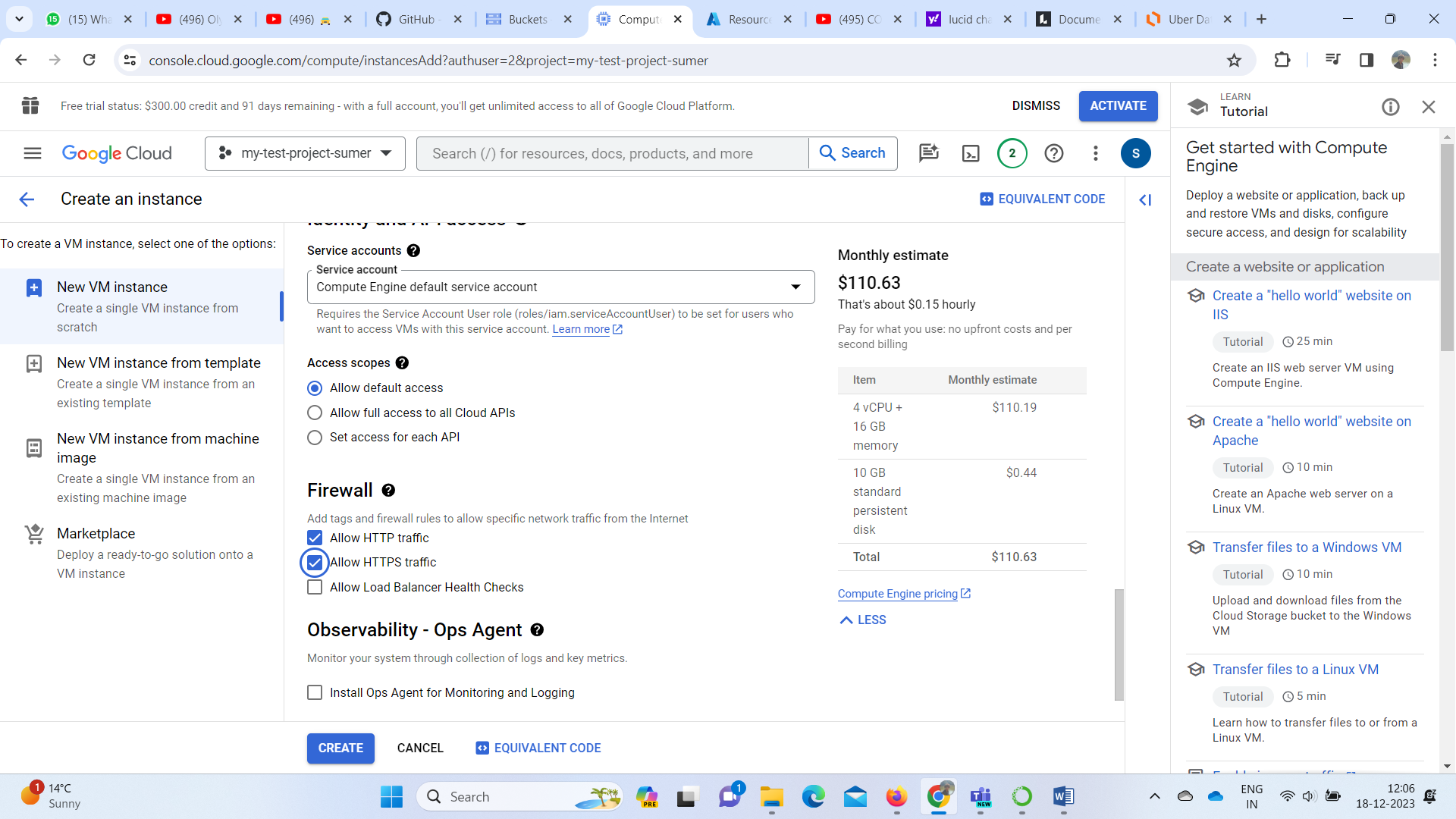
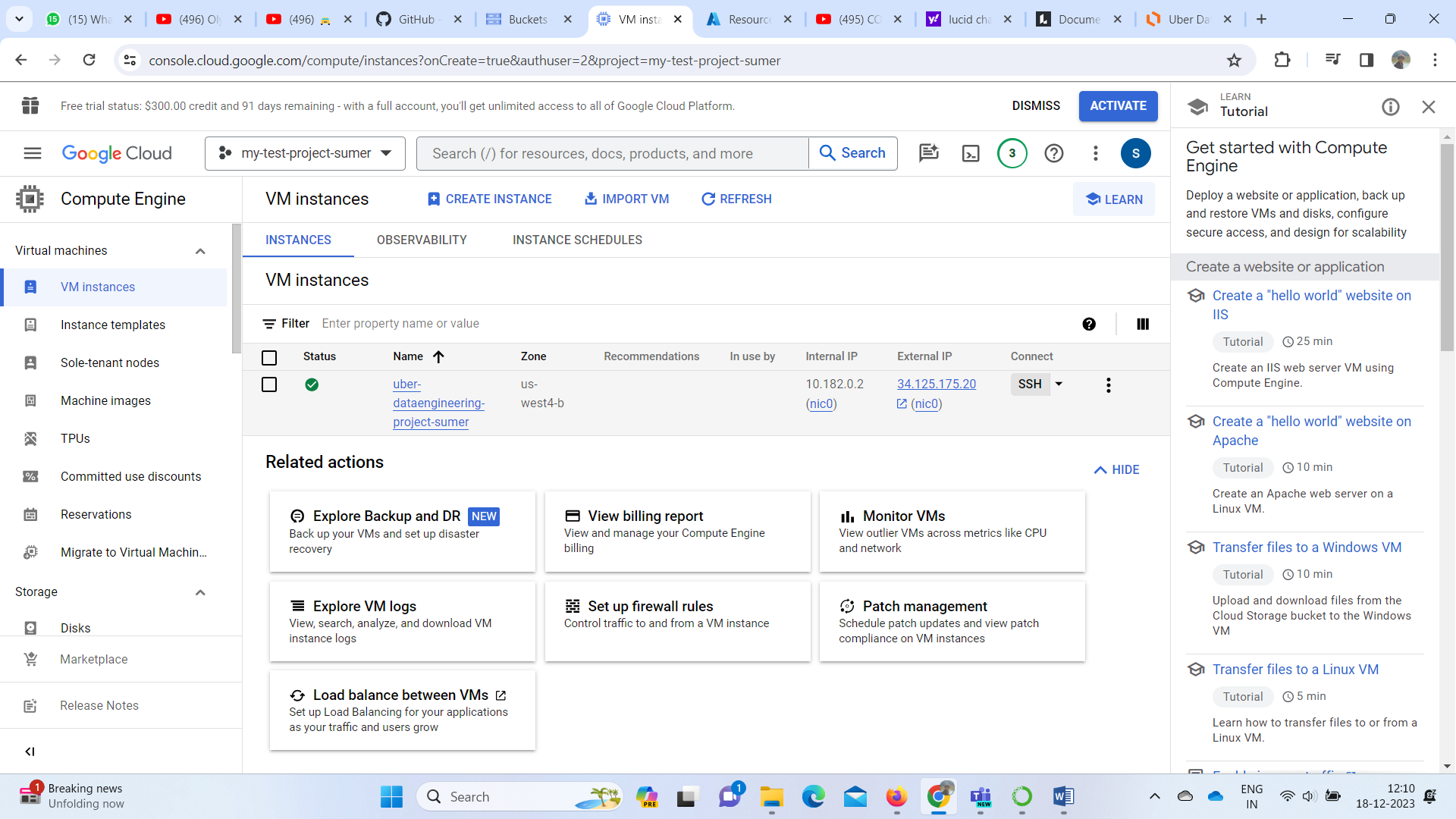
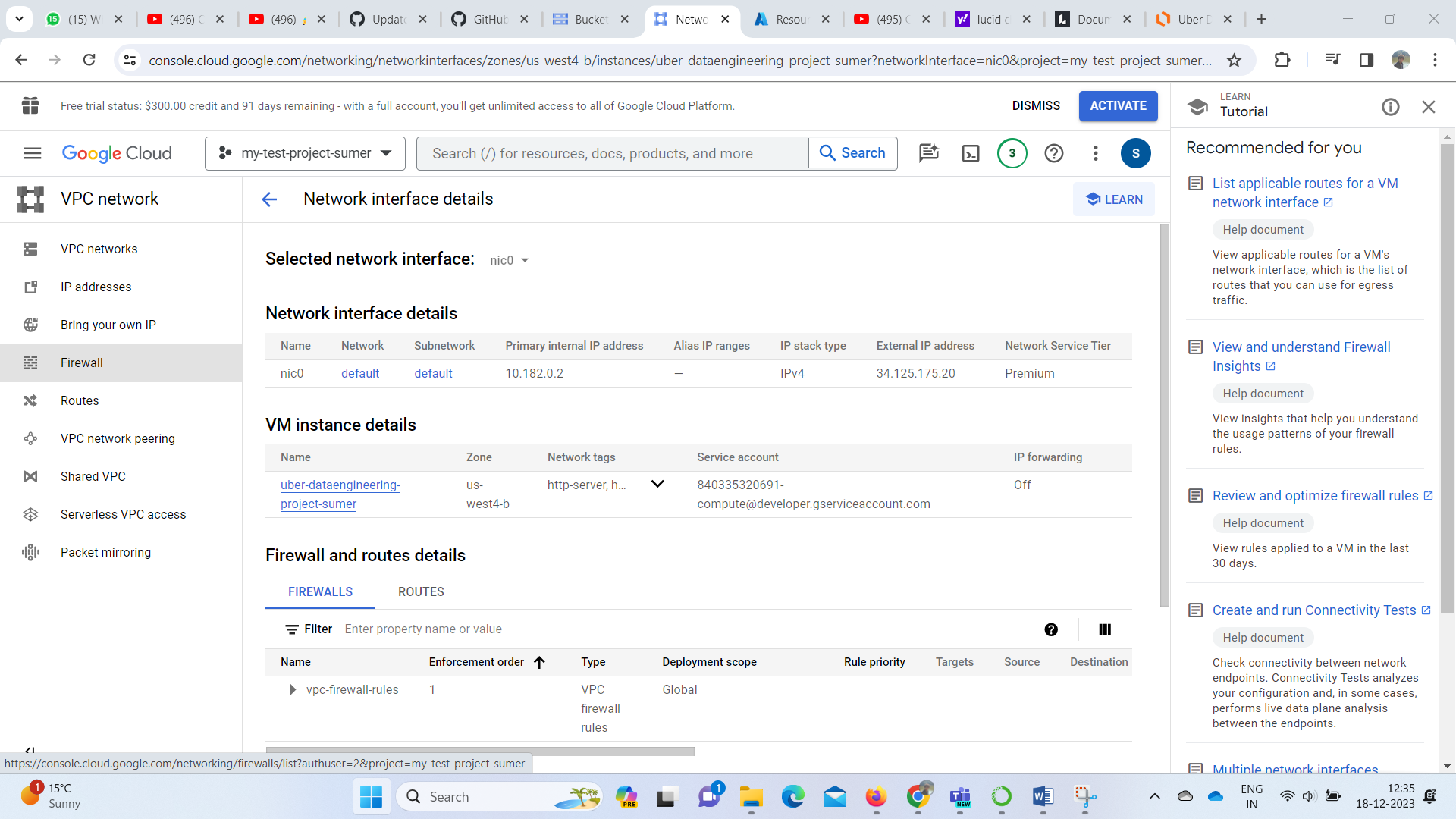
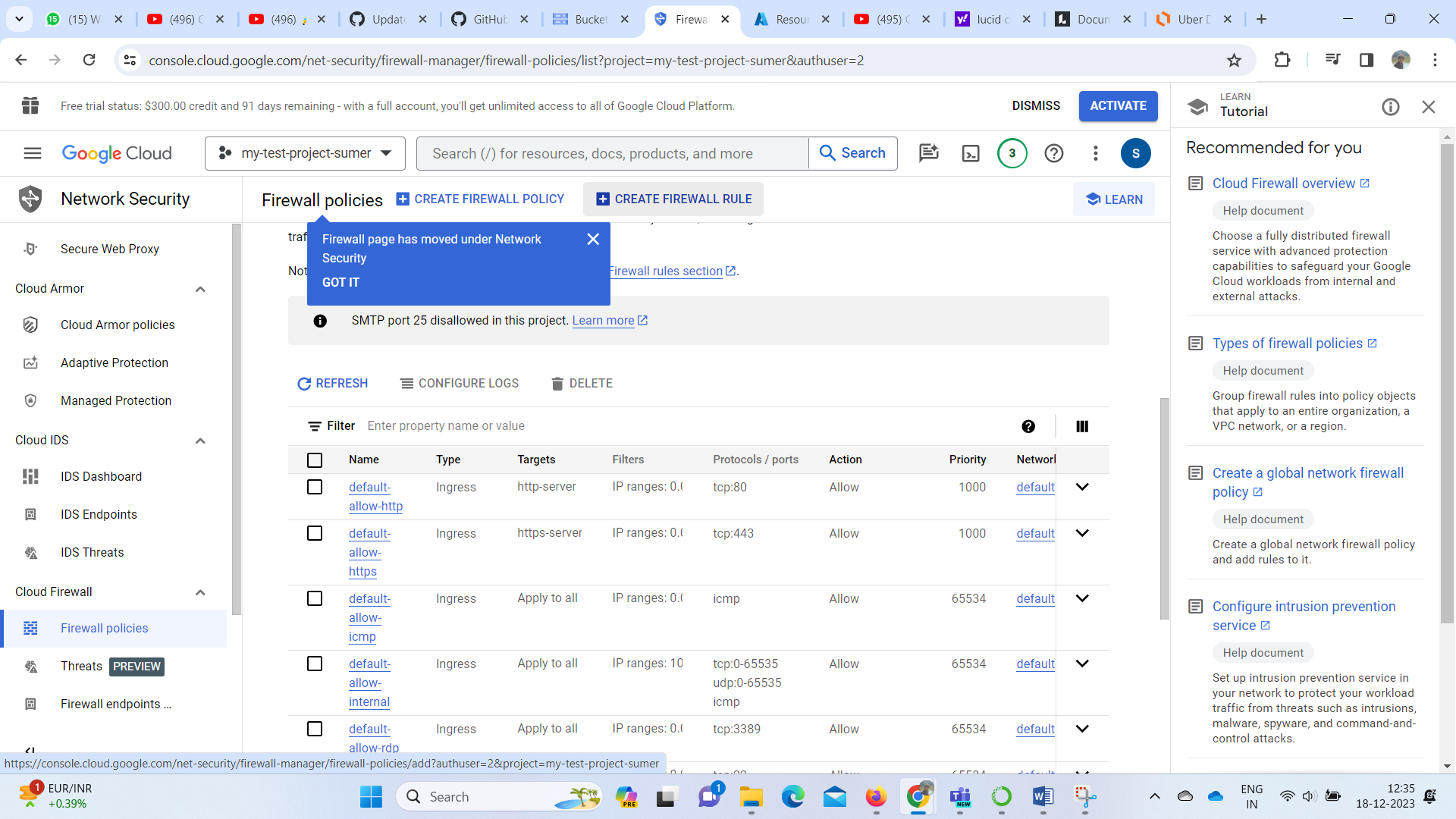
4. passenger count

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

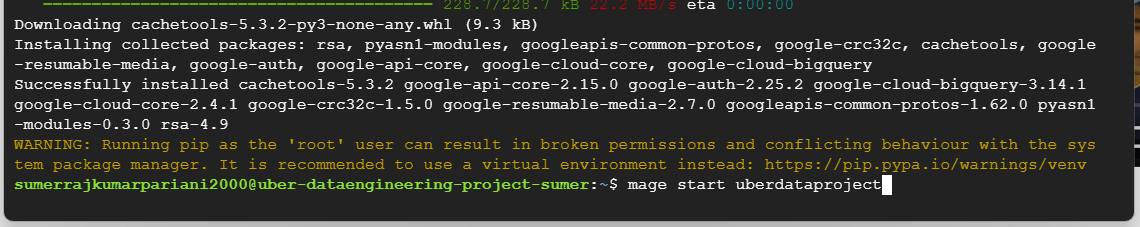
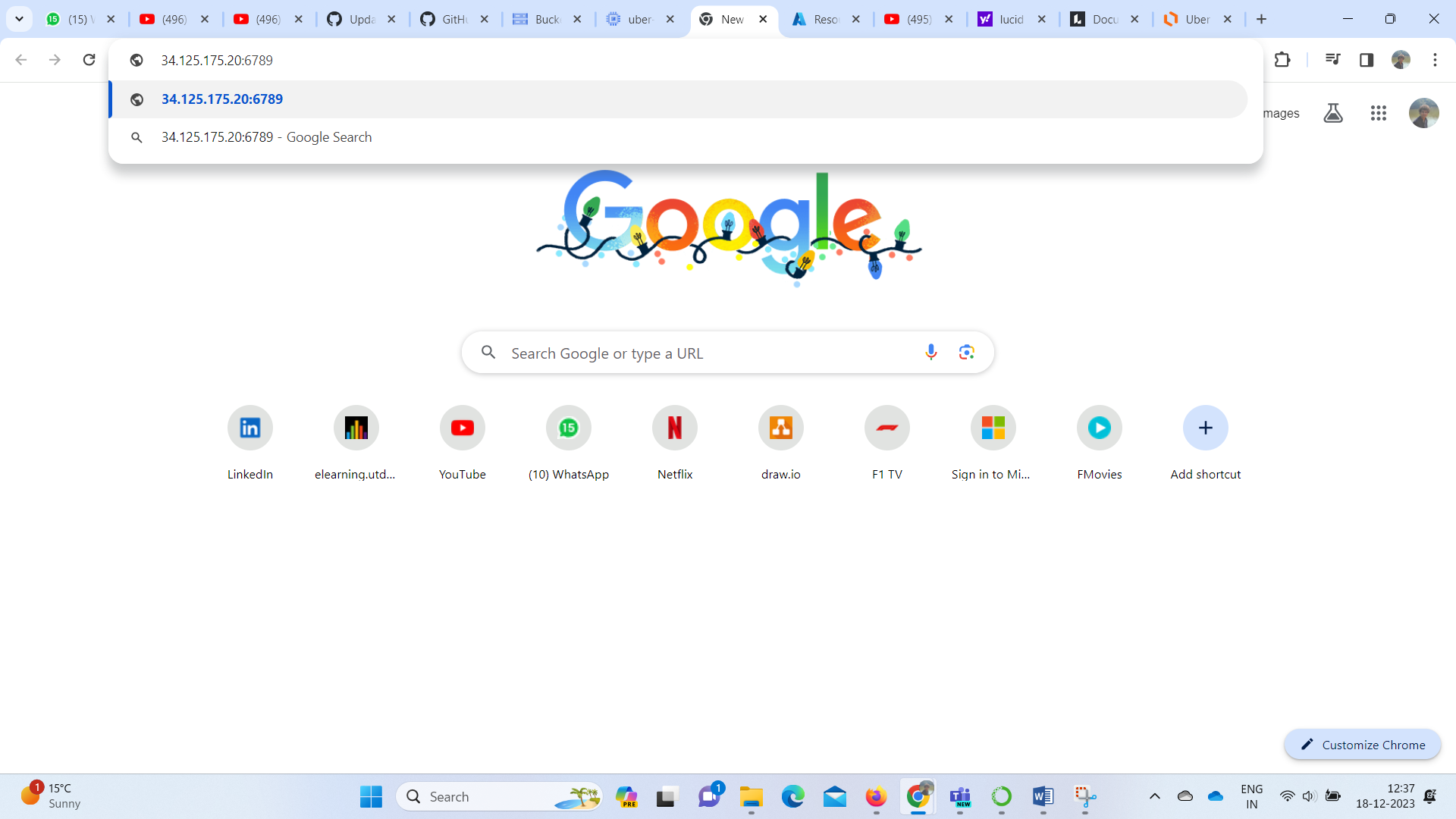
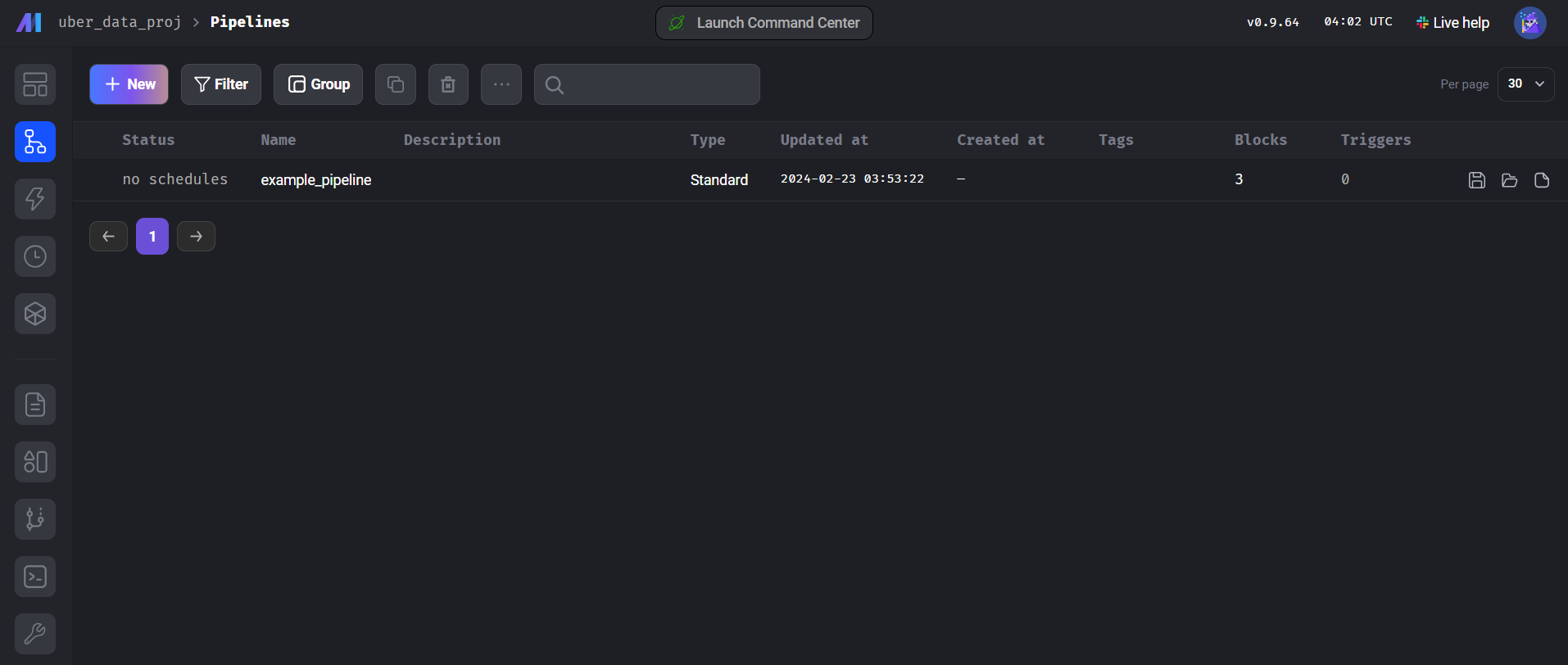
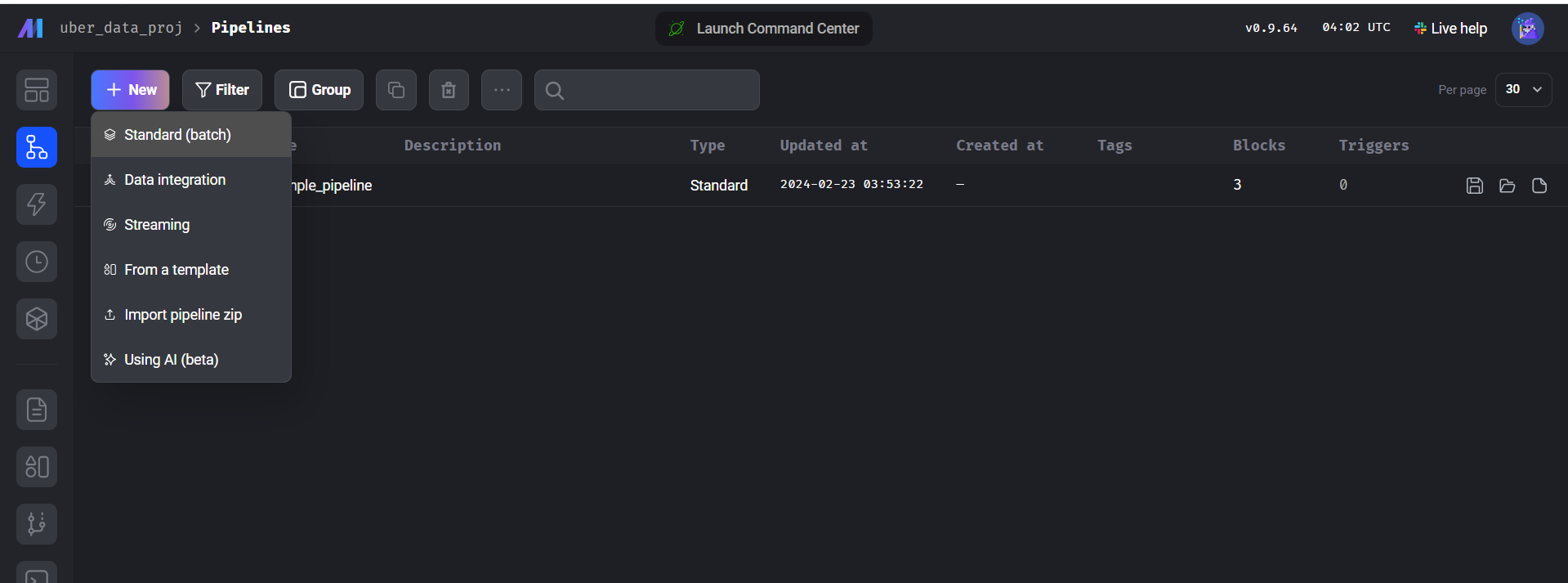
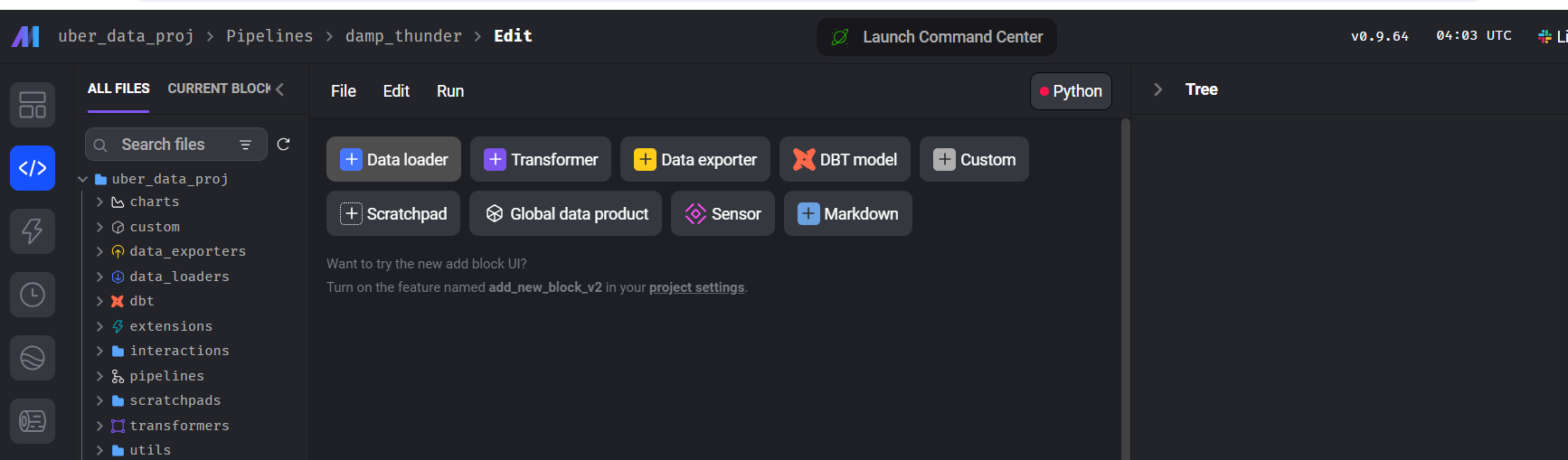
sudo pip3 install pandas

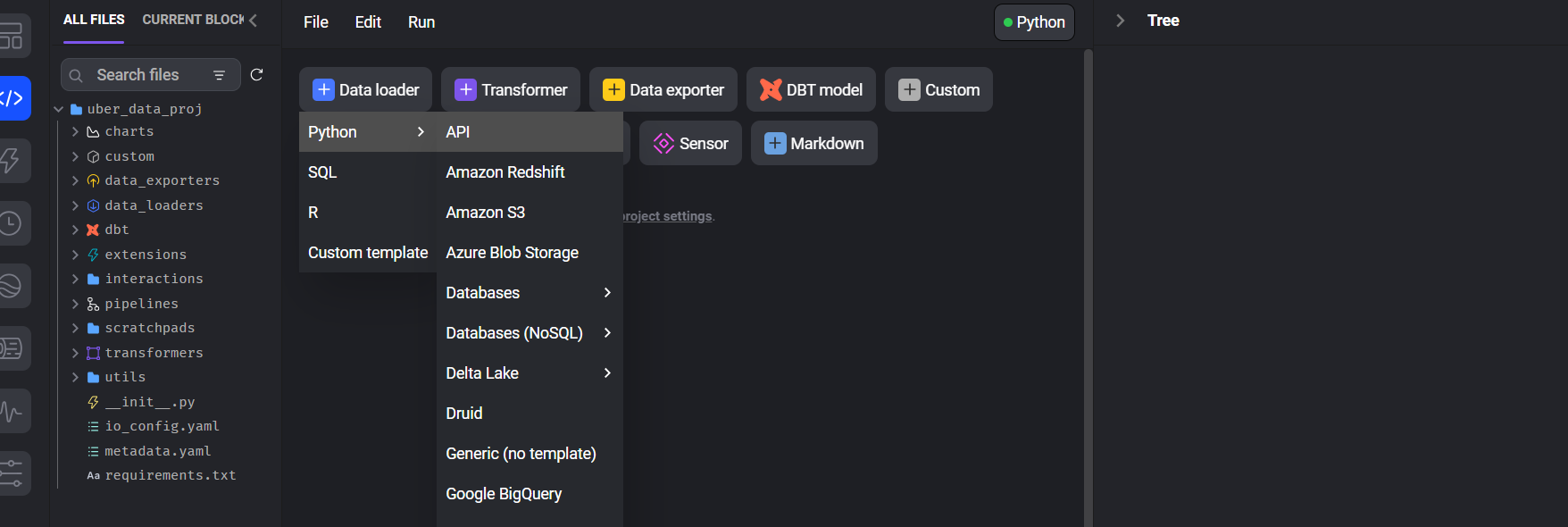
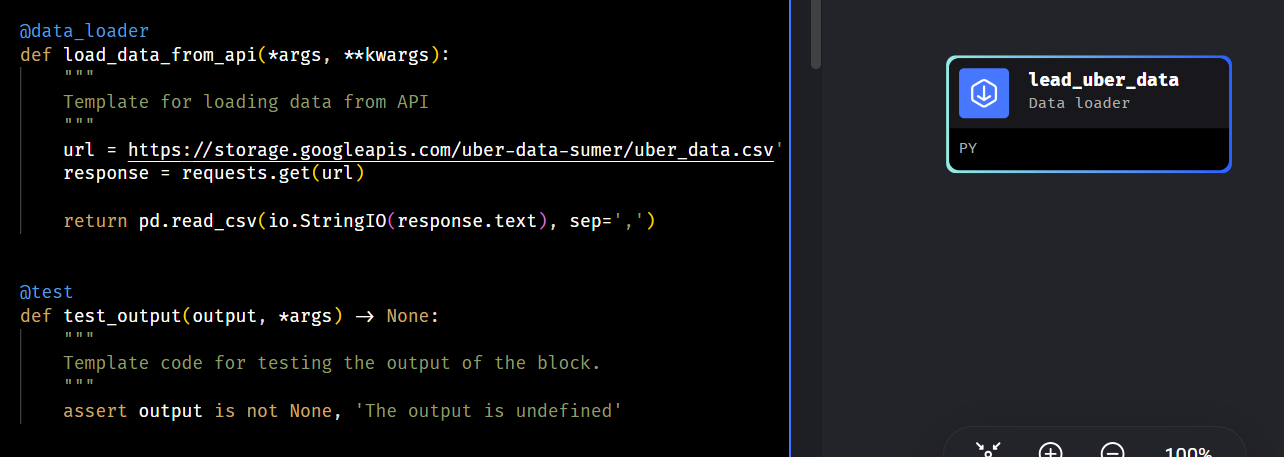
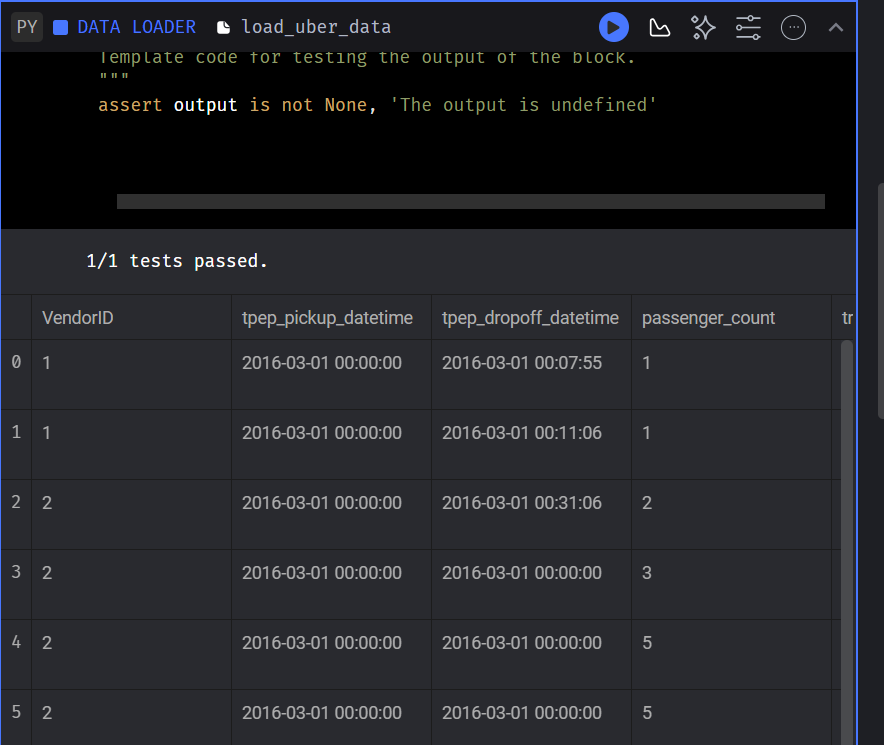
# Install Google Cloud Library

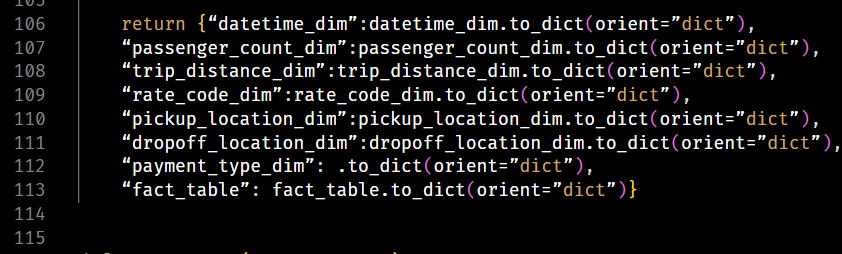
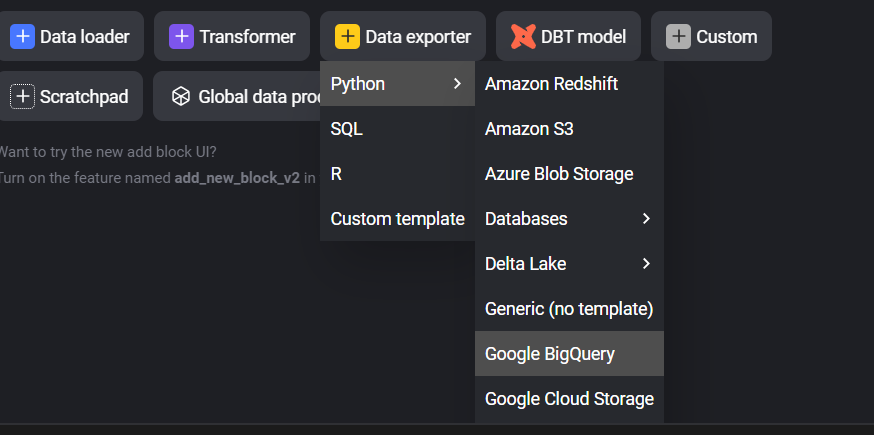
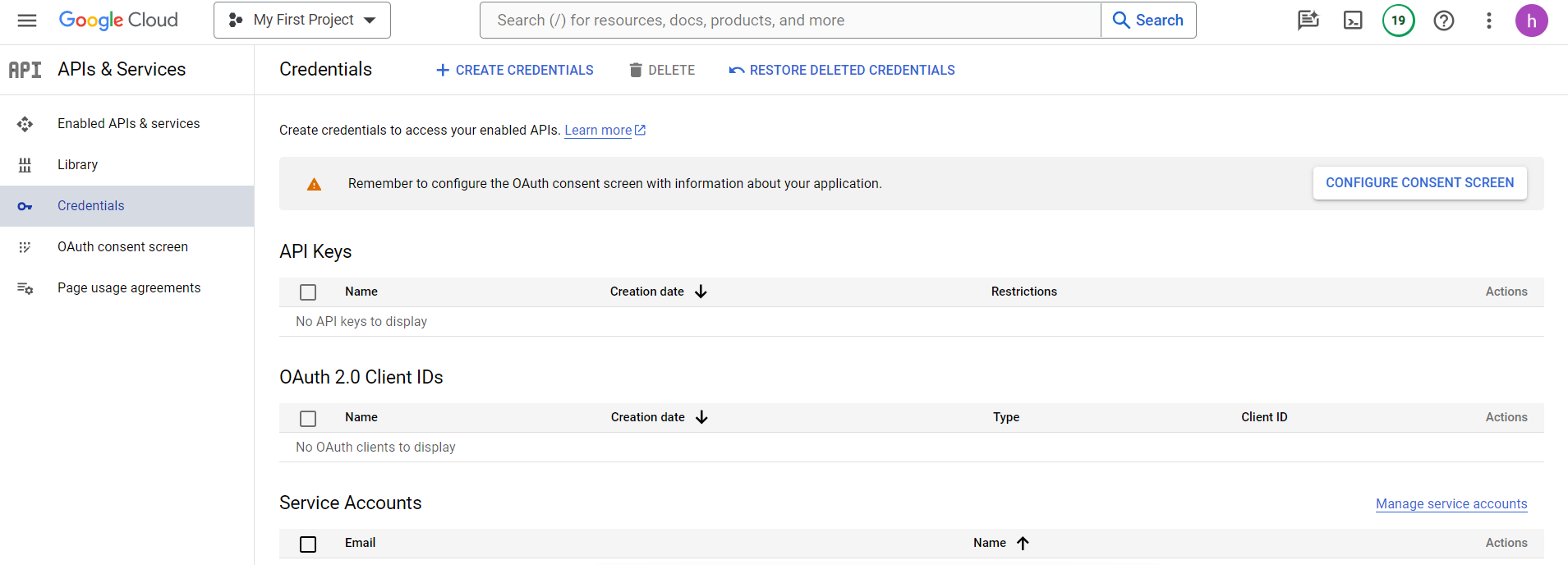
sudo pip3 install google-cloud

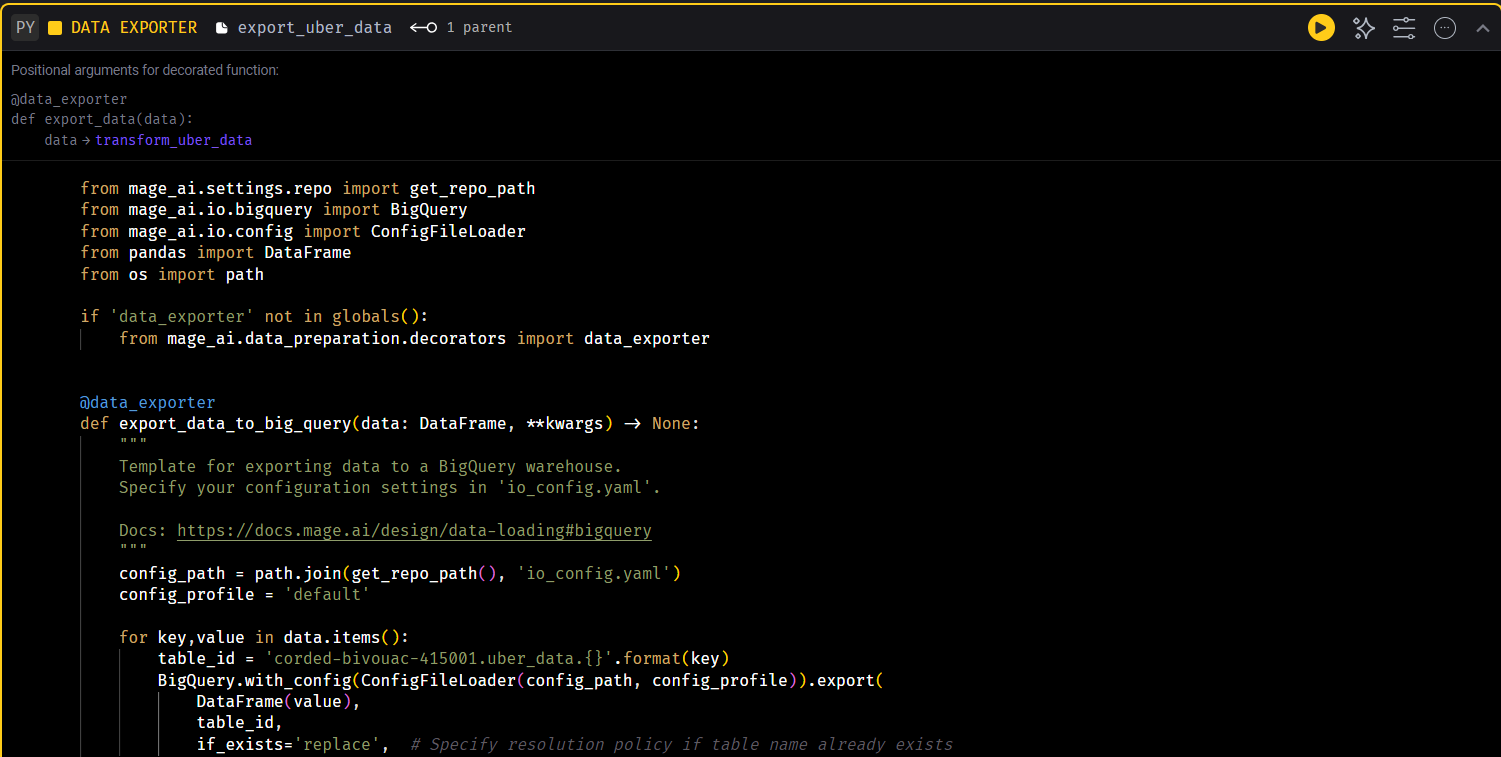
sudo pip3 install google-cloud-bigquery

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

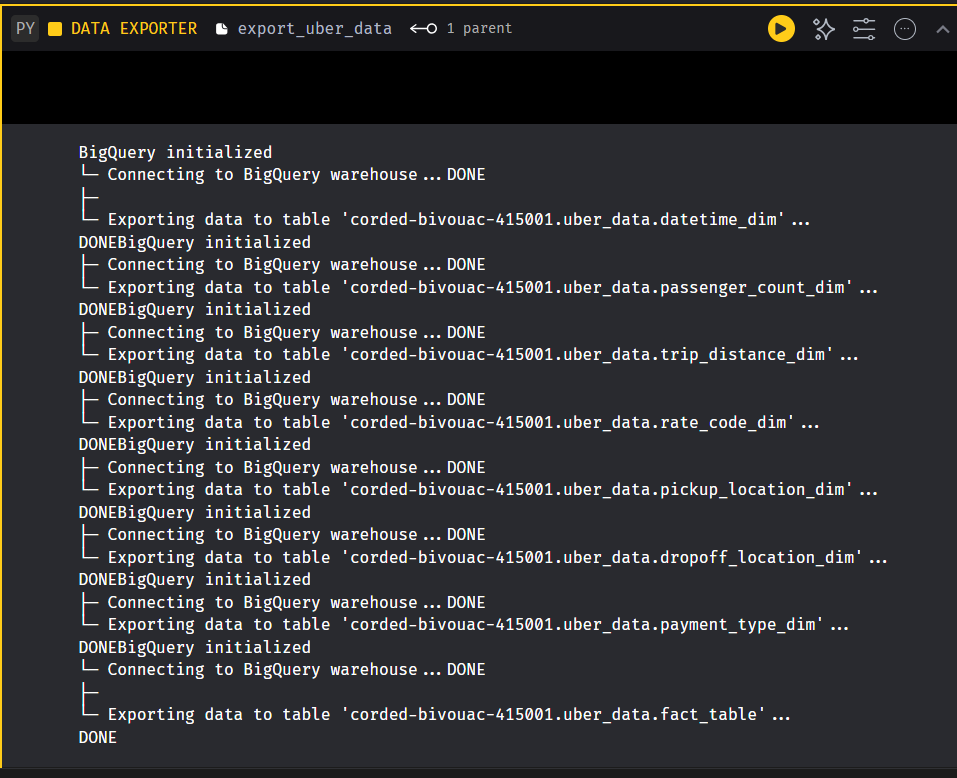
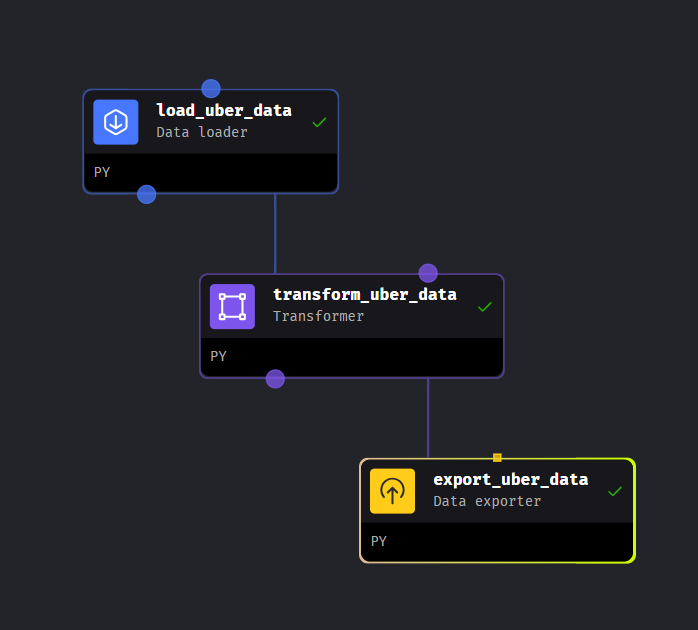
  
  
  
This will give you a port number on which the mage instance is working   
port 6789  
  
  
  
  
once we have our mage running let’s connect to mage UI .Copy the external IP of the instance and connect to the 6789 TCP port   
  
  
Here we are on our Mage UI   
  
  
  
  
Start by creating a new standard (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   
  


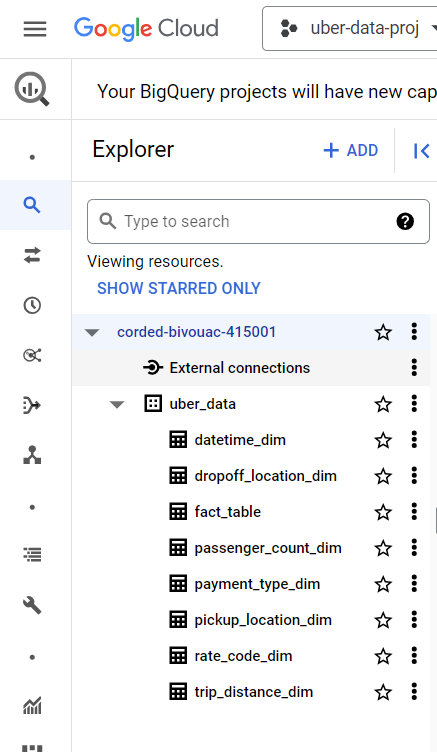
Now we move to the transformation park .Click on the transform option to open the mage code for transformation .Apply the transformation code in the section given there .Remember to return all the tables at the end .  
  
To have provide my transformation code visit this Link -<https://github.com/SumerPariani/GCP/blob/main/uber_data_analysis.ipynb>  
  
  
  
  
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   
  




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

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 .  
