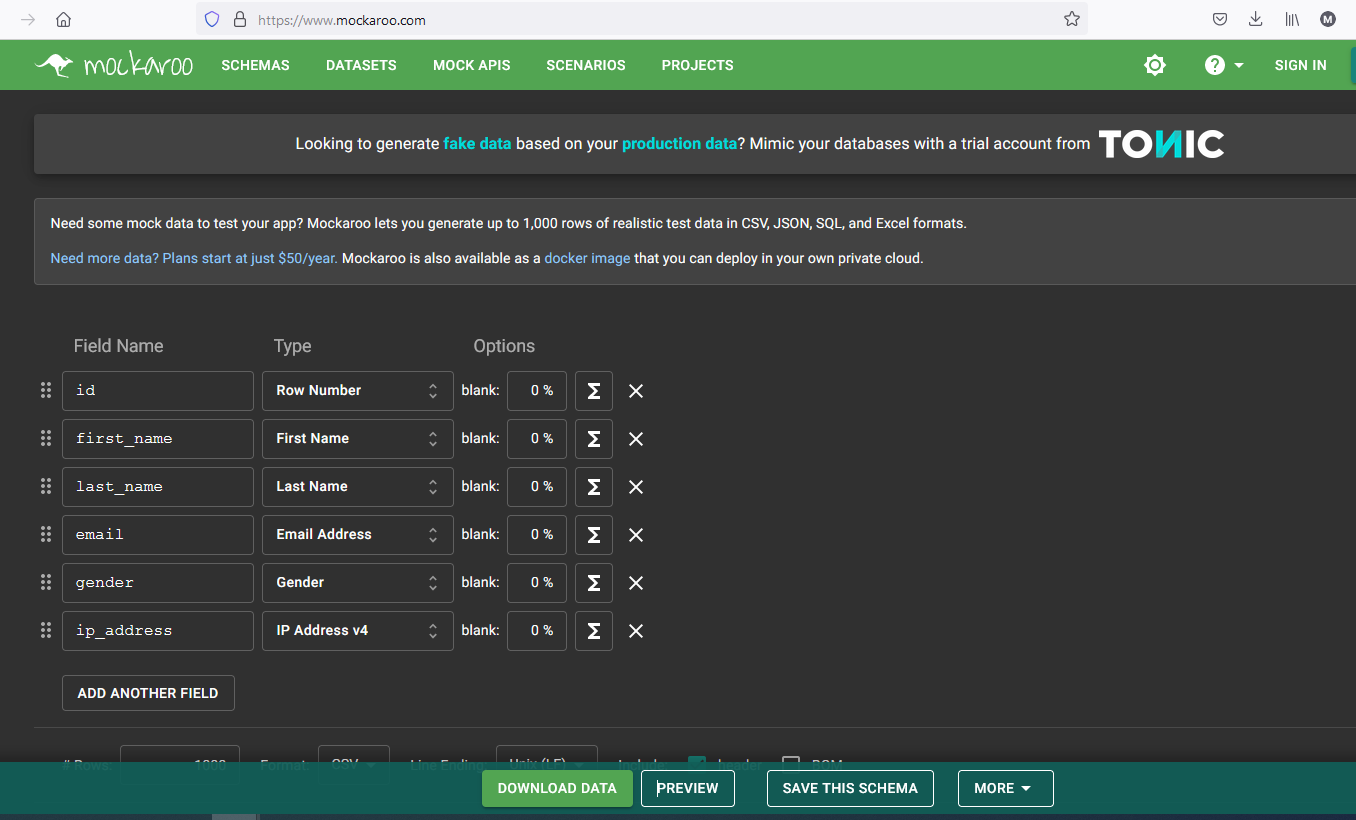
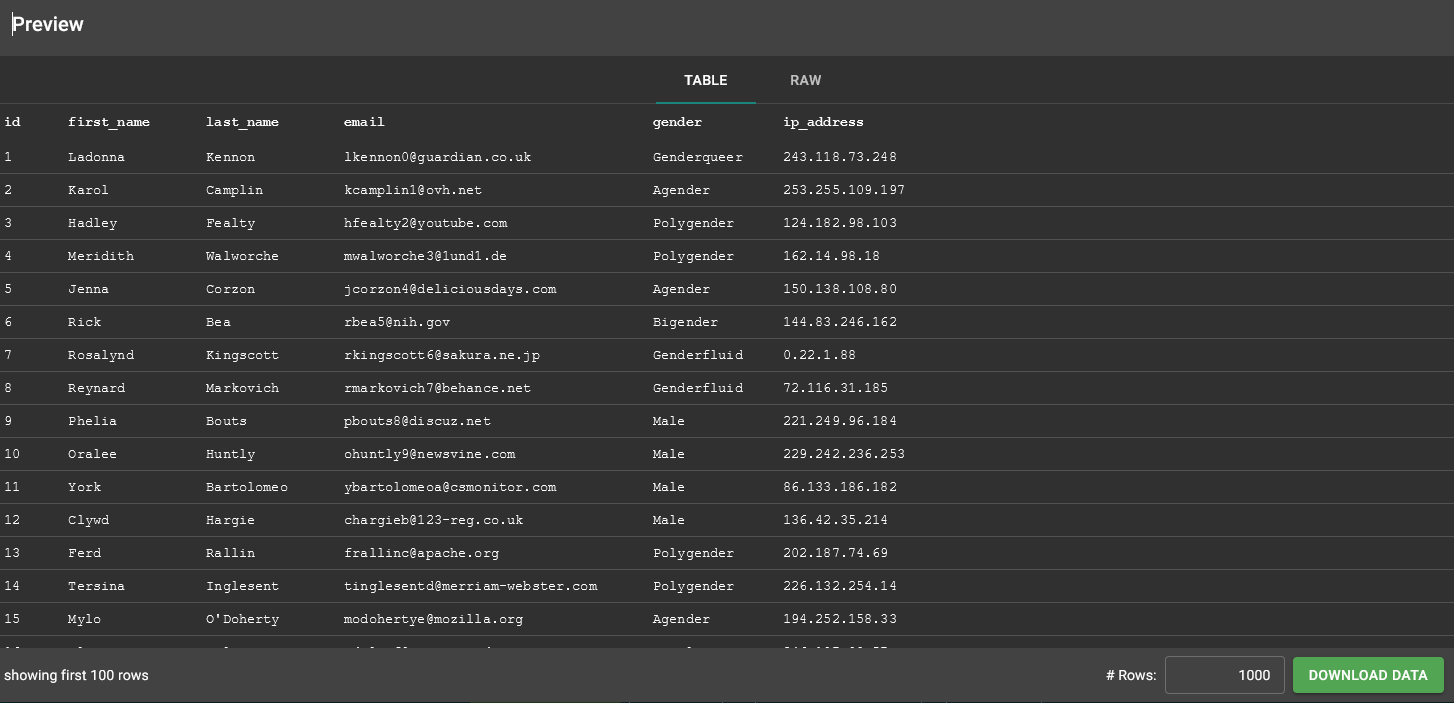
Automated load of CSV data into BigQuery via GCS bucket, Cloud Function and Python

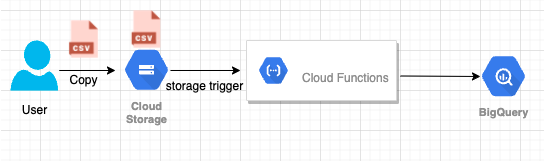
This POC is an automated loadingof CSV data into BigQuery, specifically using a Cloud Function that would automatically run whenever a new CSV file was uploaded into a Google Cloud Storage bucket.

1. **Generate a CSV file with 1000 lines of dummy data via** [**https://www.mockaroo.com/**](https://www.mockaroo.com/) **with their default schema so it looks like:**



1. **Click on PREVIEW and DOWNLOAD DATA after that and rename the csv file as testdata.csv**



****

1. **Create a main.py python file for cloud function, BigQuery Dataset, tables and storage.**

**main.py**

import os

from google.cloud import bigquery

def csv\_loader(data, context):

client = bigquery.Client()

dataset\_id = os.environ['DATASET']

dataset\_ref = client.dataset(dataset\_id)

job\_config = bigquery.LoadJobConfig()

job\_config.schema = [

bigquery.SchemaField('id', 'INTEGER'),

bigquery.SchemaField('first\_name', 'STRING'),

bigquery.SchemaField('last\_name', 'STRING'),

bigquery.SchemaField('email', 'STRING'),

bigquery.SchemaField('gender', 'STRING'),

bigquery.SchemaField('ip\_address', 'STRING')

]

job\_config.skip\_leading\_rows = 1

job\_config.source\_format = bigquery.SourceFormat.CSV

# get the URI for uploaded CSV in GCS from 'data'

uri = 'gs://' + os.environ['BUCKET'] + '/' + data['name']

# lets do this

load\_job = client.load\_table\_from\_uri(

uri,

dataset\_ref.table(os.environ['TABLE']),

job\_config=job\_config)

print('Starting job {}'.format(load\_job.job\_id))

print('Function=csv\_loader, Version=' + os.environ['VERSION'])

print('File: {}'.format(data['name']))

load\_job.result() # wait for table load to complete.

print('Job finished.')

destination\_table = client.get\_table(dataset\_ref.table(os.environ['TABLE']))

print('Loaded {} rows.'.format(destination\_table.num\_rows))

1. **Create an environment file with information about BUCKET, DATASET and TABLE**

**env.yaml**

BUCKET: csvtestbucket-301021

DATASET: csvtestdataset

TABLE: csvtable

VERSION: v14

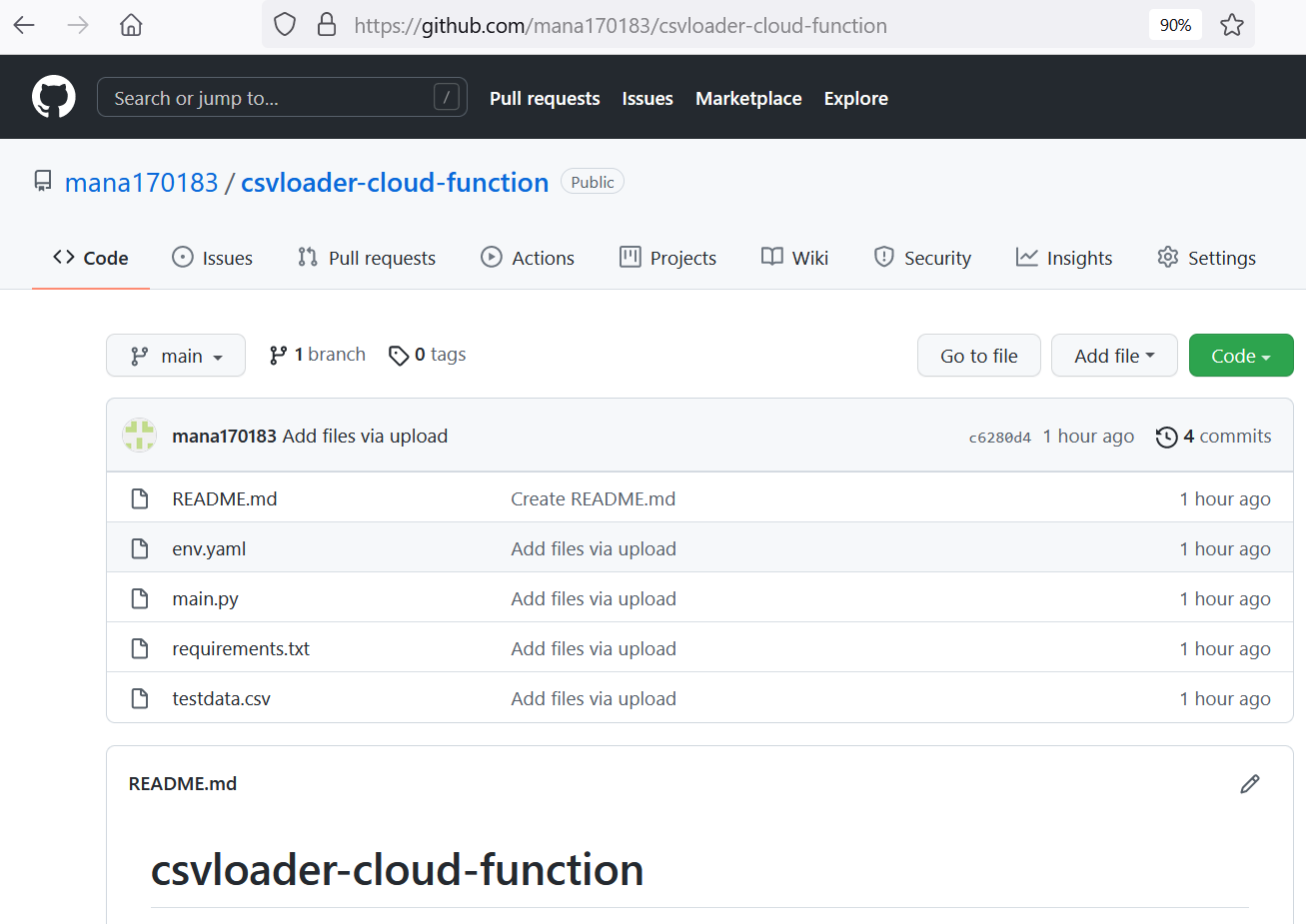
1. **Create a requirements.txt file for the necessary imports**

**requirements.txt**

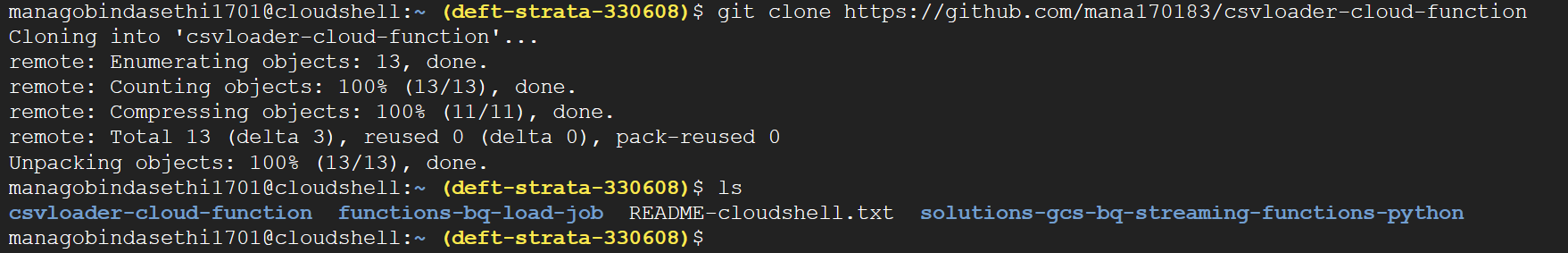
google-cloud

google-cloud-bigquery

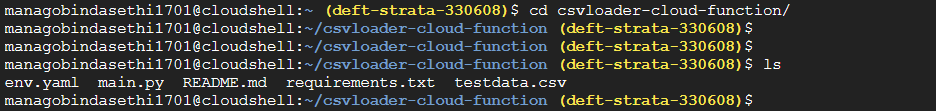
1. **Created a repository as csvloader-cloud-function on my github account with below url to clone it.**



git clone https://github.com/mana170183/csvloader-cloud-function

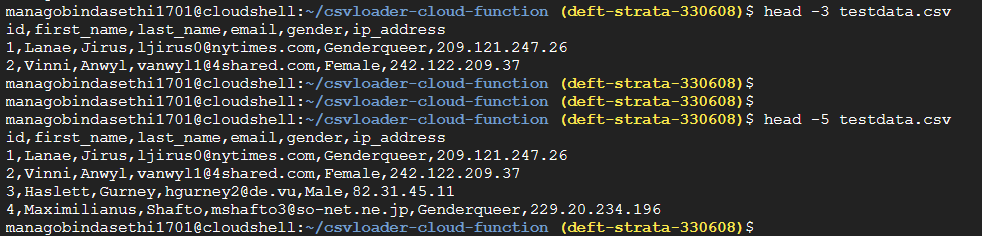


1. **After the cloning of the repository, go to the folder and type ‘ls’ to check for all files**



1. **Check for testdata.csv contents by typing below command:**

head -3 testdata.csv



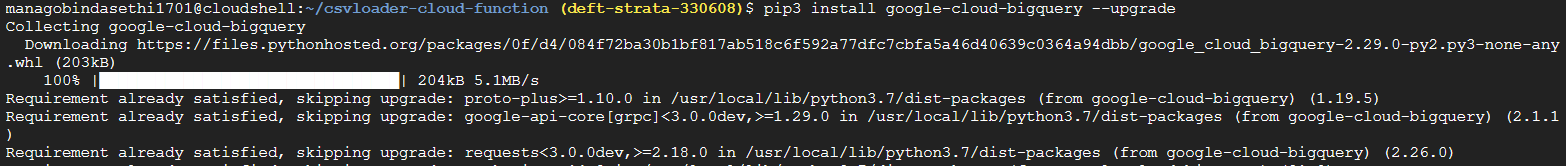
1. **Create a Google Cloud Storage bucket as below command:**

gsutil mb gs://csvtestbucket-301021



1. **Install the necessary python dependencies and packages:**

pip3 install google-cloud-bigquery --upgrade



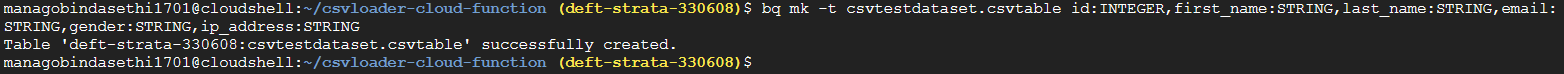
1. **Create the BigQuery Dataset by mentioning the GCP project name:**

bq mk --dataset deft-strata-330608:csvtestdataset



1. **Create the table on the dataset:**

bq mk -t csvtestdataset.csvtable id:INTEGER,first\_name:STRING,last\_name:STRING,email:STRING,gender:STRING,ip\_address:STRING



**Alternately, create a schema.json file as below and deploy the command as below to load the table’s schema.**

bq mk csvtestdataset.csvtable schema.json

schema.json

[

{

"name": "id",

"type": "INTEGER",

"mode": "NULLABLE"

},

{

"name": "first\_name",

"type": "STRING",

"mode": "NULLABLE"

},

{

"name": "last\_name",

"type": "STRING",

"mode": "NULLABLE"

},

{

"name": "email",

"type": "STRING",

"mode": "NULLABLE"

},

{

"name": "gender",

"type": "STRING",

"mode": "NULLABLE"

},

{

"name": "ip\_address",

"type": "STRING",

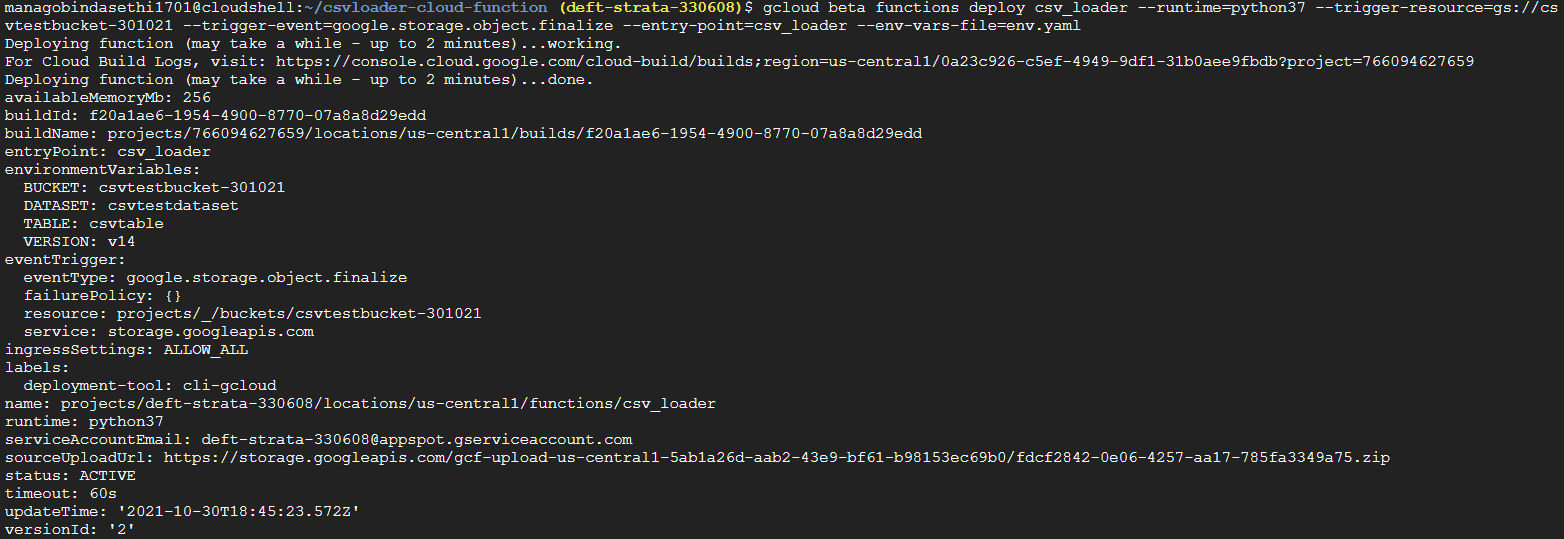
"mode": "NULLABLE"

}

]

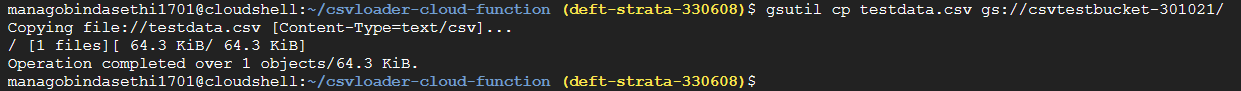
1. **Deploy the cloud function giving all details as below: we will add a trigger on the storage bucket that will fire every time a new file is added to the bucket. here we are creating a cloud function named “csv\_loader”:**

gcloud beta functions deploy csv\_loader --runtime=python37 --trigger-resource=gs://csvtestbucket-301021 --trigger-event=google.storage.object.finalize --entry-point=csv\_loader --env-vars-file=env.yaml



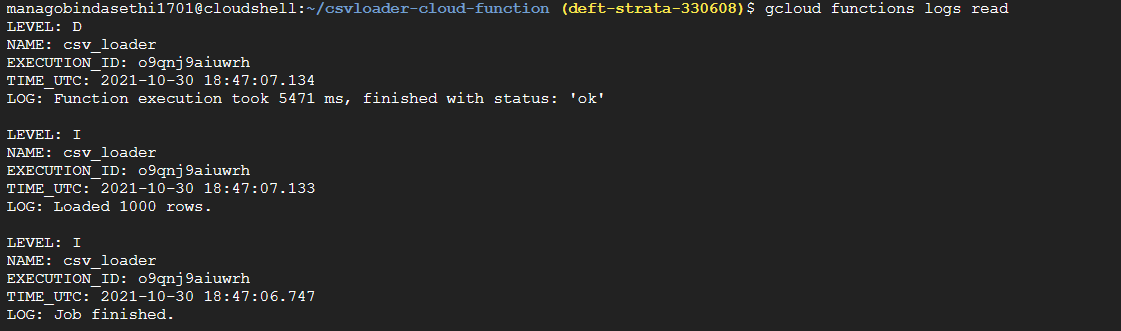
1. **After the function is deployed, yay! copy your test data CSV into the bucket:**

gsutil cp testdata.csv gs://csvtestbucket-301021/



1. Now that we have copied the CSV file in the bucket, the function should fire! check the cloud function logs:

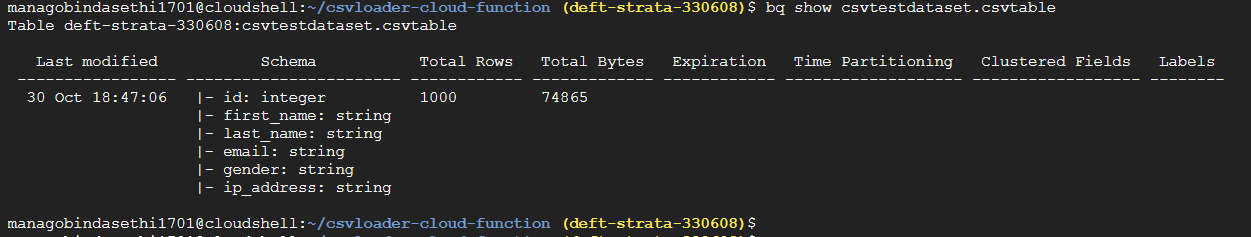
gcloud functions logs read



looks like the function ran as expected!

1. **Check the BigQuery again and see if the row count has changed.**

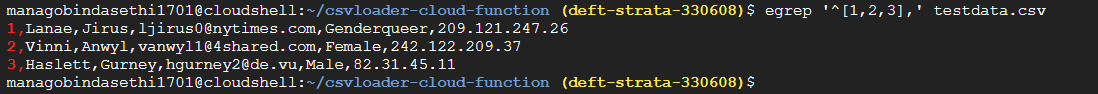
bq show csvtestdataset.csvtable

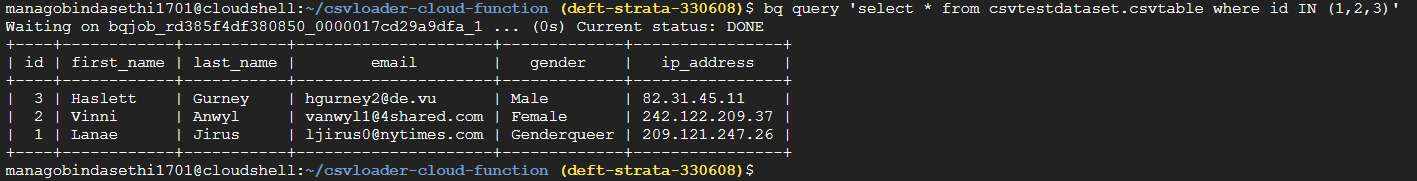


great! there are now 1000 rows. looking good.

1. **As a final check, lets compare the first 3 rows of the CSV with the first 3 rows of BigQuery table**

egrep '^[1,2,3],' testdata.csv

bq query 'select \* from csvtestdataset.csvtable where id IN (1,2,3)'



1. **Run the BigQuery command to check the data on table:**

SELECT \* FROM `deft-strata-330608.csvtestdataset.csvtable` LIMIT 1000

