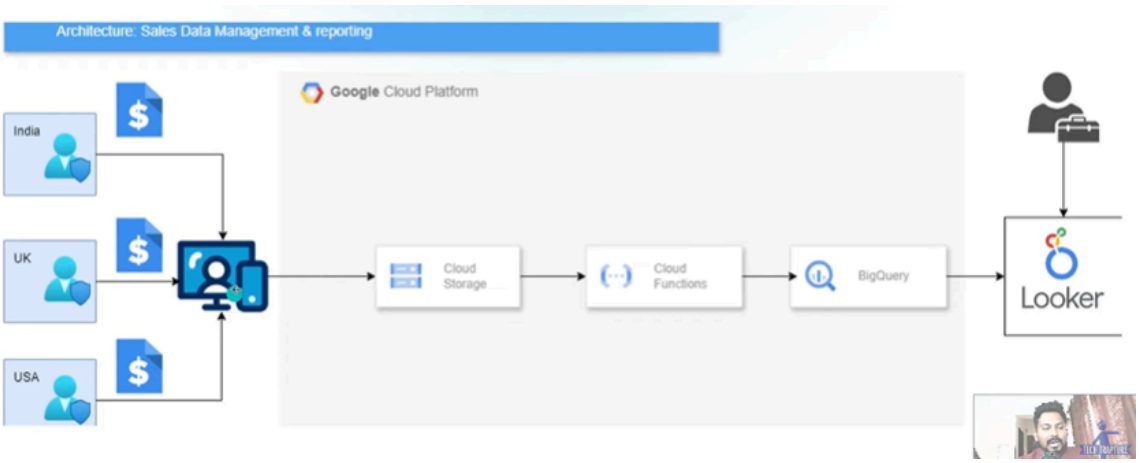
Project Requiremnt

1. 

2. So here we'll create a webportal…where shop managers can upload their shop's sales data
3. And we'll extract this data into big query and transform the data
4. Later we develop dashboards/views to present it to CEO

Technical architecture

1. 

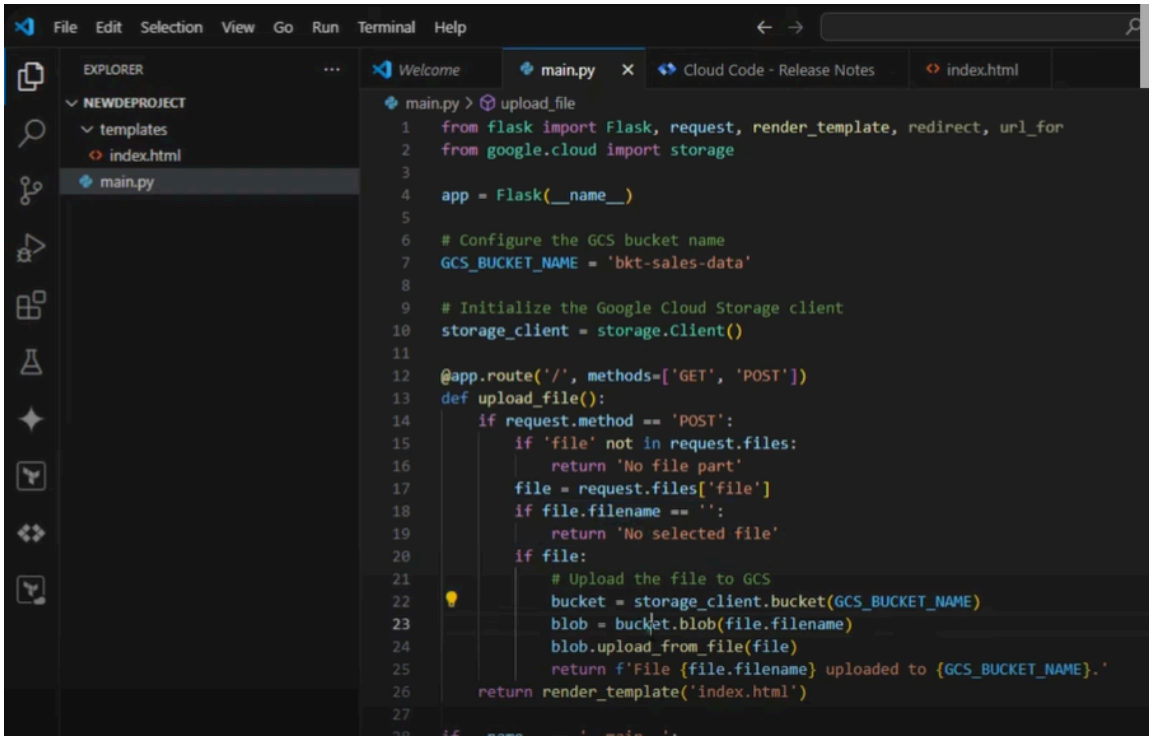
2. Here we get the files(CSV's) from our portal…and we store the files in GCS storage bucket

3. Then using cloud functions such as DataFlow/DataFusion or cloud composer…we can load the data into BigQiery
4. Here we'll use cloud function to migrate the data from GCS to GBQ
5. We'll trigger the cloud function whenever there is a file in GCS….and we'll write python code …to detect the file and load the file into bigquery
6. So we have 3 csv files…and we create a portal and upload them in the portal

Practical

1. So here we have created a flast webapplication where we can upload files..
2. The uploaded files must be stored in GCS storage bucket
3. For this we will create a bucket…and integrate that with our flask application
4. Here we have choose a file and uploaded in the site

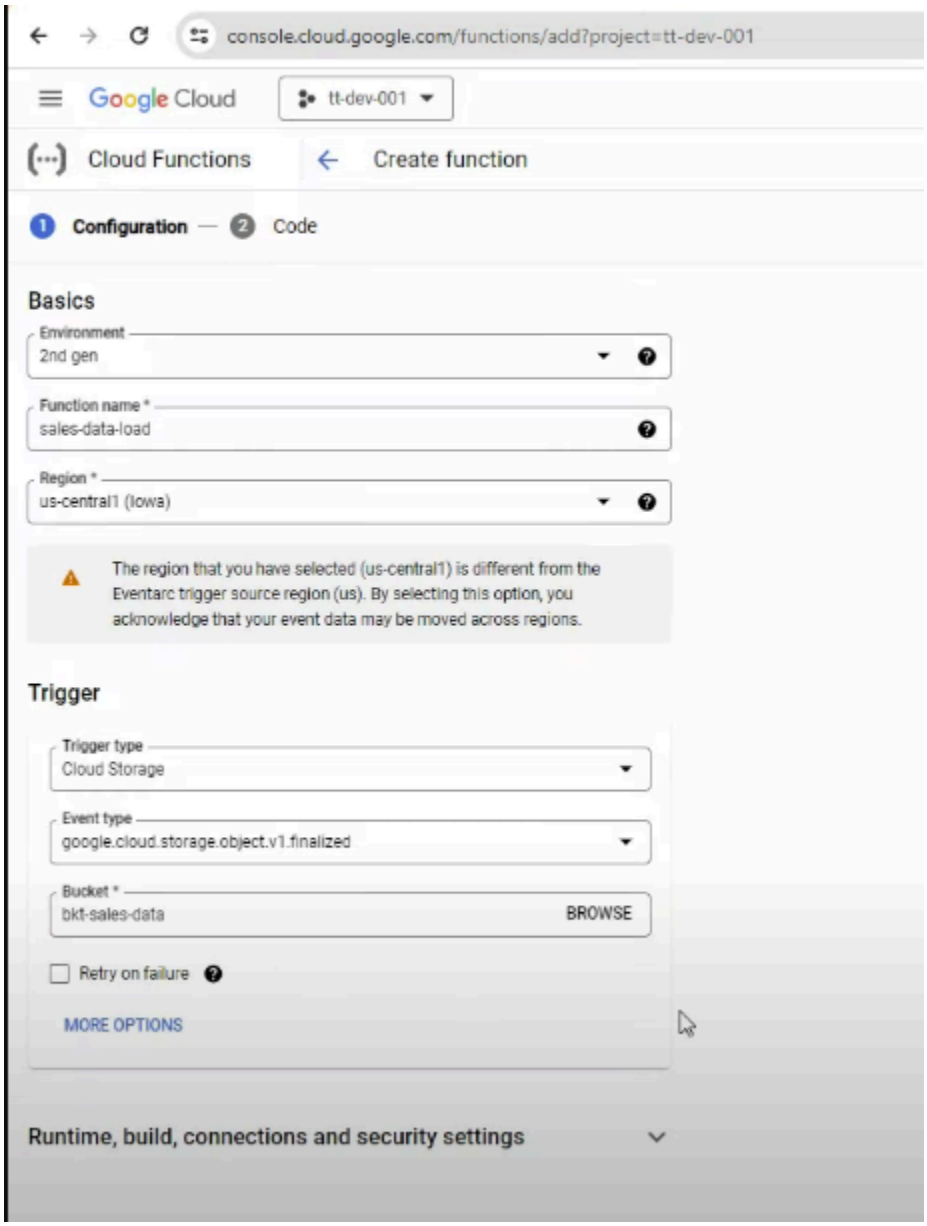
5. Now here we can see our file is available in our Bucket that we have integrated
6. Here we have uploaded a file using POST request



7. So when we upload files..they are stored in this bucket

Cloud Function

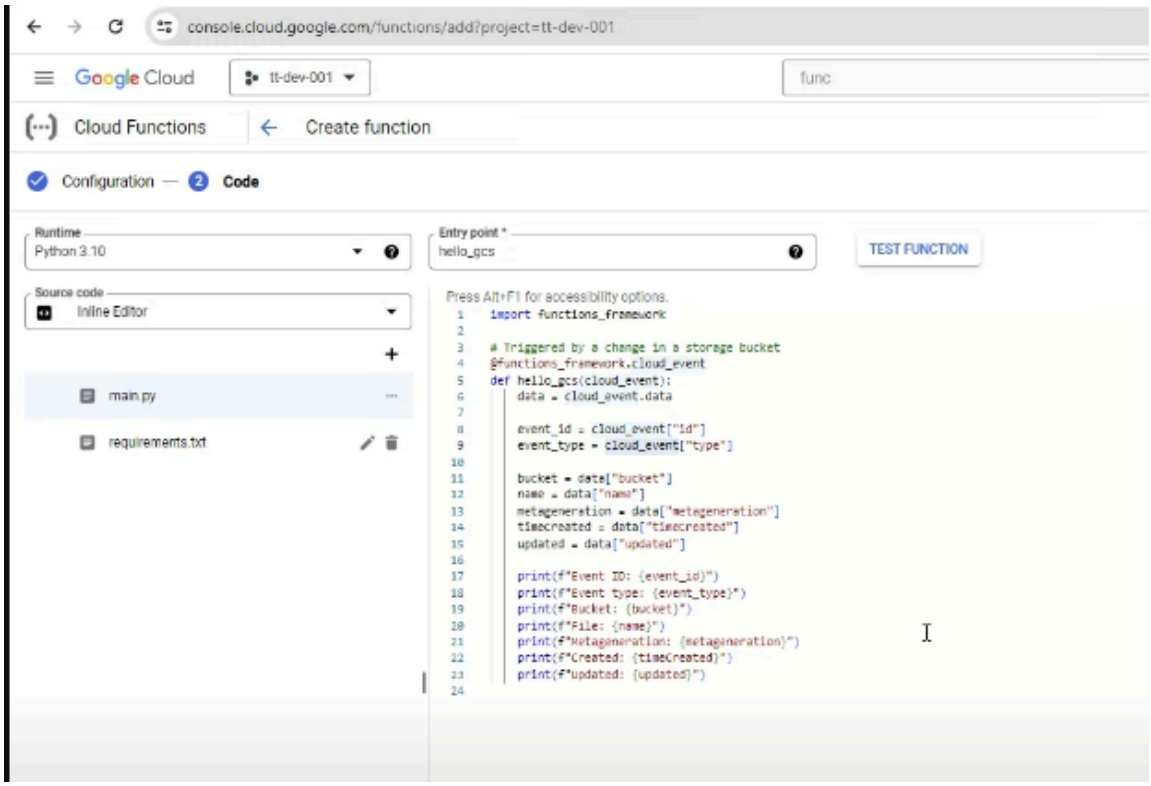
1. Lets open a cloud function and create one
2. So here we'll be creating a cloud function in 2nd gen env as it has more features



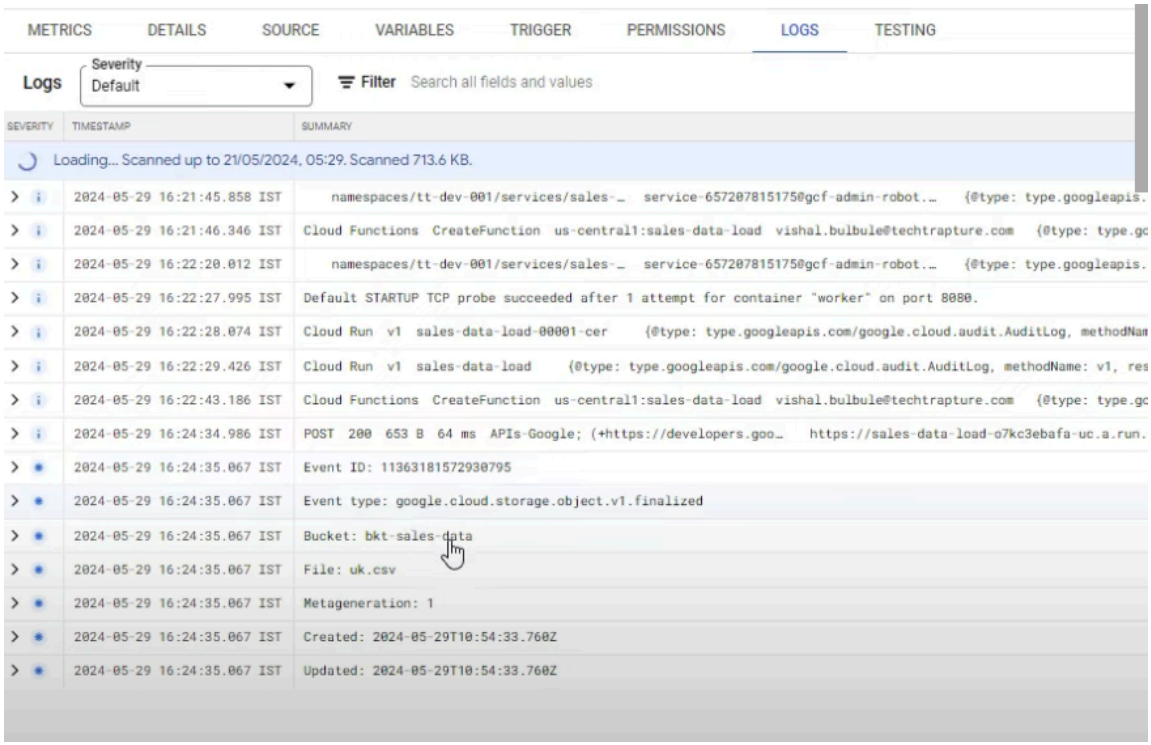
we will trigger this function..when there is file in our GCS bucket

3. Now we have to write a python script which extracts data from GCS storage bucket and loads it in GBQ

4. Here first we'll get an auto generated code..which gives us info about file,bucket name etc



5. Here we'll upload the file..and check our cloud function..whether its getting trigger or not



we can see its working without any issue

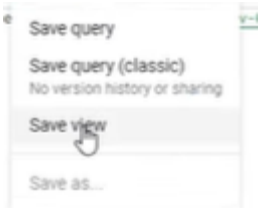
6. Here in this code load_bq function gets the files from GCS and store it in GBQ

```
@functions_framework.cloud_event
def hello_gcs(cloud_event):
    data = cloud_event.data

    event_id = cloud_event["id"]
    event_type = cloud_event["type"]

    bucket = data["bucket"]
    filename = data["name"]
    metageneration = data["metageneration"]
    timeCreated = data["timeCreated"]
    updated = data["updated"]

    print(f"Event ID: {event_id}")
    print(f"Event type: {event_type}")
    print(f"Bucket: {bucket}")
    print(f"File: {filename}")
    print(f"Metageneration: {metageneration}")
    print(f"Created: {timeCreated}")
    print(f"Updated: {updated}")
    load_bq(filename)


dataset = 'sales'
table = 'orders'
def load_bq(filename):
    client = bigquery.Client()
    filename = filename
    table_ref = client.dataset(dataset).table(table)
    job_config = LoadJobConfig()
    job_config.source_format = bigquery.SourceFormat.CSV
    job_config.skip_leading_rows = 1
    job_config.autodetect = True
    uri = f'gs://bkt-sales-data/{filename}'
    load_job = client.load_table_from_uri(uri, table_ref, job_config=job_config)
    load_job.result()
    time.sleep(10)
    num_rows = load_job.output_rows
    print(f"{num_rows} rows loaded into {table}.")
```

7. Next we'll deploy this function,,,and upload a file for testing purpose
8. Later we'll upload the biggest file 1000k rows

Views

1. After uploading the data to bigqiery…we can create views…



2. To create a view..click on save…and create aview
3. Next we will create dashboard based on these data..using looker studio
4.