

Assessment 2: GDDA707 - Advanced Data Engineering

Project 2: Data Integration

Mira Torririt

GDDA 707

Lecturer: Dr Zarah Zandi

School of Technology
Graduate Diploma in Data Analytics (Level 7)

Section 1: Summary of the project

This assessment aims to show the integration process and make the data more accessible and reliable for business strategic planning and data analysis.

Most businesses have numerous business applications accessed by employees, such as CRM, online ordering systems, product inventory, logistics, and accounting databases. Data integration connects all these various sources in one storage.

Data integration is the process of combining data from different data sources into a unified form. This assessment started with data input going to the database (as a data source), which is then extracted, transformed, and loaded (ETL) generally to the data warehouse (as a unified storage).

In Part A, the two datasets obtained from the Kaggle website were placed in two cloud buckets using the dataflow for ETL to BigQuery as the destination. These data were cleaned before loading into the database to ensure data integrity and prepare them for future analysis. Dataflow was chosen as a methodology for ETL. During this process, many errors occurred due to connections and configurations, which will be discussed explicitly in the next section.

In part B, the datasets used were also downloaded from the Kaggle website; both are customer transaction data. This time, the methodology used to store it in the Cloud platform is the data ingestion pipeline. The datasets were initially stored in one GCP bucket and eventually transferred to the Hadoop Distributed File System (HDFS), which is widely used for big data processing and analysis. Just like in Part A, errors occurred during the application of the said methodology; one of these is the creation of the destination folder in HDFS.

Section 2: Task-Specific Description and Technical Details

Part A: ETL Operations

Task 1: Selection of Datasets and Tools

The scenario involves integrating two e-commerce sales databases into the cloud platform. Two datasets obtained from the Kaggle website were chosen for the scenario.

The first one is the Amazon Sales Report, which consists of the following data: index, order ID, date, status, fulfillment, sales channel, ship-service-level, style, SKU, category, size, ASIN, courier status, quantity, currency, amount, ship-city, ship-state, ship-postal-code, ship-country, promotion-ids, fulfilled-by, unnamed: 22. (The Devastator, (n.d.)).

The second dataset concerns a one-year business transaction for a UK-based shop. Since 2007, the shop has been selling online gifts and homewares for adults and children. Customers worldwide can order directly from their website. They also have small business owner customers who buy through their retail outlet channels. The dataset comprises more than 500k rows and eight columns for the transaction no., date, product no, product, price, quantity, customer no., and country (Ramos, G., (n.d.)).

The datasets are also available in my github account: <https://github.com/Myres16/Data-Analytics-Assessments/tree/main/707>

The detailed tools used in the data integration are as follows:

Data Integration Tools		
Dataset	Amazon Sale Report	Sales Transactions v.4a
Data cleaning and uploading to the database	Python	Python
Database / Storage	MongoDB	GCP Bucket
ETL (Extract, Transform, and Load)	Dataflow	Dataflow
Data warehouse	Big Query	Big Query
Cloud platform	Google Cloud Platform	Google Cloud Platform

The reasons for using different storage are:

1. MongoDB has a flexible schema, making storing and retrieving data easier.
2. To access the CSV file directly, I manually uploaded it to the Google Cloud Platform (GCP) Bucket to utilize it.
3. To explore the MongoDB and GCP Bucket functions.

I chose GCP because the credits were available. To maximize its use, I chose the dataflow, which is free of charge in GCP and is a product of Google, which is responsible for the ETL process.

Here's the process to access the dataflow:

1. Go to the GCP navigation menu, scroll down, click MORE PRODUCTS, scroll down, look for ANALYTICS, and click Dataflow.
2. Go to Jobs and click [+ CREAT JOB FROM TEMPLATE](#)
3. Fill out the Create job from the template

Important: The regional endpoint should match the source database's regional settings (for example, MongoDB Atlas) and the destination database's (for example, the Big Query). The dataflow template should match the source and destination databases (for example, MongoDB to BigQuery)

4. Once you have filled out the dataflow template, it will show you the required parameters to be filled out.
5. Click RUN JOB.

Task 2: Load and Pre-processing

I downloaded the datasets from the Kaggle website and imported them to Python using the pandas library and pd for pre-integration. read function.

- Amazon Sales Dataset

		index	Order ID	Date	Status	Fulfilment	Sales Channel	ship-service-level	Style	SKU	Category	...	currency	Amount	ship-city	ship-state								
0	0	405-5731545	8079784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389-KR-NP-S	Set ...	INR	647.62	MUMBAI	MAHARASHTRA										
1	1	171-1101146	9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXL	kurta ...	INR	406.00	BENGALURU	KARNATAKA									
2	2	404-7273146	0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	kurta ...	INR	329.00	NAVI MUMBAI	MAHARASHTRA									
3	3	403-8133951	9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	Western Dress ...	INR	753.33	PUDUCHERRY	PUDUCHERY									
4	4	407-7240320	1069790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXL	Top ...	INR	574.00	CHENNAI	TAMIL NADU									
128970	128970	406-7673107	6001380-7673107	05-31-22	Shipped	Amazon	Amazon.in	Expedited	JNE3697	JNE3697-KR-XL	kurta ...	INR	517.00	HYDERABAD	TELANGANA									
128971	128971	402-7544318	9551604-7544318	05-31-22	Shipped	Amazon	Amazon.in	Expedited	SET401	SET401-KR-NP-M	Set ...	INR	999.00	GURUGRAM	HARYANA									
128972	128972	407-3152358	9547469-3152358	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0157	J0157-DR-XXL	Western Dress ...	INR	690.00	HYDERABAD	TELANGANA									
128973	128973	402-0545956	6184140-0545956	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0012	J0012-SKD-XS	Set ...	INR	1199.00	Halol	Gujarat									
128974	128974	408-8728312	7436540-8728312	05-31-22	Shipped	Amazon	Amazon.in	Expedited	J0003	J0003-SET-S	Set ...	INR	696.00	Raipur	CHHATTISGARH									

128975 rows x 24 columns

Data was cleaned to prepare it for integration and future analysis. To facilitate, I used the isnull function to look for null values and thefillna function to replace the columns with null values. I saved the cleaned CSV file, as this is the one to be imported into the database.

```
AmazonDF.isnull().sum()
```

index	0
Order ID	0
Date	0
Status	0
Fulfilment	0
Sales Channel	0
ship-service-level	0
Style	0
SKU	0
Category	0
Size	0
ASIN	0
Courier Status	6872
Qty	0
currency	7795
Amount	7795
ship-city	33
ship-state	33
ship-postal-code	33
ship-country	33
promotion-ids	49153
B2B	0
fulfilled-by	89698
Unnamed: 22	49050
dtype: int64	

```
AmazonDF['Unnamed: 22'].fillna('TRUE', inplace=True)
```

```
AmazonDF['Courier Status'].fillna('Cancelled', inplace=True)
```

```
AmazonDF['currency'].fillna('INR', inplace=True)
```

```
AmazonDF['Amount'].fillna(0.00, inplace=True)
```

```
AmazonDF['fulfilled-by'].fillna('FedEx', inplace=True)
```

```
AmazonDF['ship-city'].fillna('SOUTH', inplace=True)
```

```
AmazonDF['ship-state'].fillna('DELHI', inplace=True)
```

```
AmazonDF['ship-postal-code'].fillna('110015', inplace=True)
```

```
AmazonDF['ship-country'].fillna('IN', inplace=True)
```

```
AmazonDF['promotion-ids'].fillna('NotAvailable', inplace=True)
```

```
AmazonDF.isnull().sum()
```

index	0
Order ID	0
Date	0
Status	0
Fulfilment	0
Sales Channel	0
ship-service-level	0
Style	0
SKU	0
Category	0
Size	0
ASIN	0
Courier Status	0
Qty	0
currency	0
Amount	0
ship-city	0
ship-state	0
ship-postal-code	0
ship-country	0
promotion-ids	0
B2B	0
fulfilled-by	0
Unnamed: 22	0
dtype: int64	

```
#Saving to a new CSV file
AmazonDF.to_csv('AmazonDF.csv')
```

- Sales Transaction Dataset

```
import pandas as pd

SalesTransactionDF=pd.read_csv('Sales_Transaction_v.4a.csv')
SalesTransactionDF
```

	TransactionNo	Date	ProductNo	ProductName	Price	Quantity	CustomerNo	Country
0	581482	12/9/2019	22485	Set Of 2 Wooden Market Crates	21.47	12	17490.0	United Kingdom
1	581475	12/9/2019	22596	Christmas Star Wish List Chalkboard	10.65	36	13069.0	United Kingdom
2	581475	12/9/2019	23235	Storage Tin Vintage Leaf	11.53	12	13069.0	United Kingdom
3	581475	12/9/2019	23272	Tree T-Light Holder Willie Winkie	10.65	12	13069.0	United Kingdom
4	581475	12/9/2019	23239	Set Of 4 Knick Knack Tins Poppies	11.94	6	13069.0	United Kingdom
...
536345	C536548	12/1/2018	22168	Organiser Wood Antique White	18.96	-2	12472.0	Germany
536346	C536548	12/1/2018	21218	Red Spotty Biscuit Tin	14.09	-3	12472.0	Germany
536347	C536548	12/1/2018	20957	Porcelain Hanging Bell Small	11.74	-1	12472.0	Germany
536348	C536548	12/1/2018	22580	Advent Calendar Gingham Sack	16.35	-4	12472.0	Germany
536349	C536548	12/1/2018	22767	Triple Photo Frame Cornice	20.45	-2	12472.0	Germany

536350 rows × 8 columns

```
SalesTransactionDF.isnull().sum()

TransactionNo      0
Date              0
ProductNo        0
ProductName      0
Price             0
Quantity          0
CustomerNo       55
Country           0
dtype: int64
```

```
SalesTransactionDF['CustomerNo'].fillna('12835', inplace=True)
```

```
SalesTransactionDF.isnull().sum()

TransactionNo      0
Date              0
ProductNo        0
ProductName      0
Price             0
Quantity          0
CustomerNo       0
Country           0
dtype: int64
```

```
SalesTransactionDF.to_csv('Cleaned_SalesTransactionDF.csv')
```

Task 3: ETL Operations and Integration

Amazon Sales Dataset – From Python to MongoDB

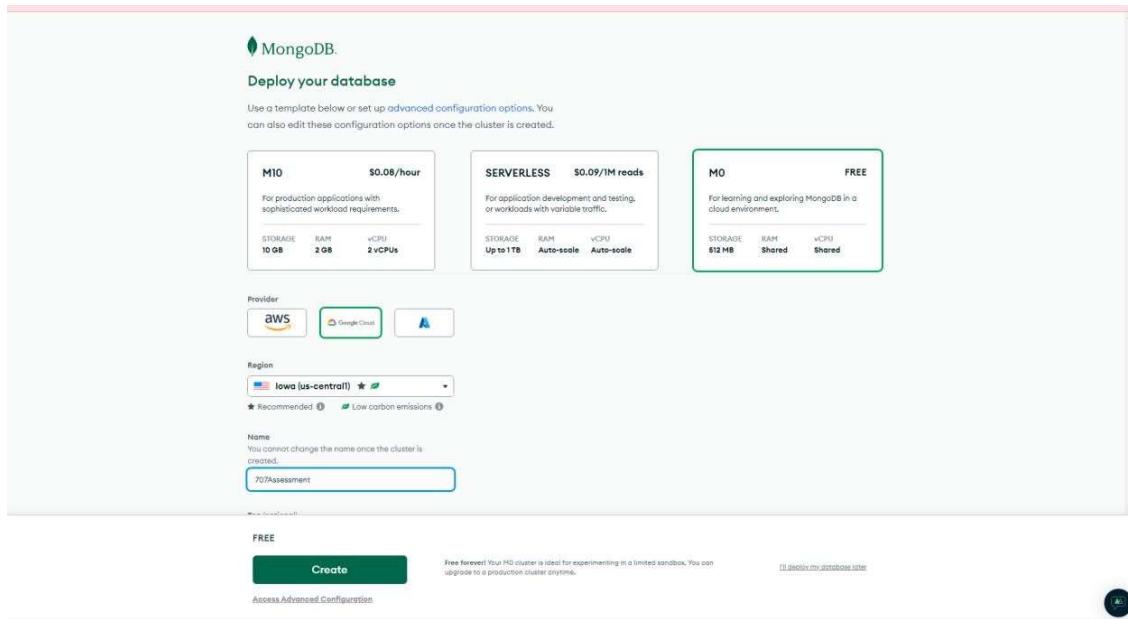
- To set up MongoDB Atlas, go to the Overview page.

The screenshot shows the MongoDB Atlas Overview page. On the left, there's a sidebar with navigation links like Overview, Deployment, Database, Data Lake, Services, Device Sync, Triggers, Data API, Data Federation, Atlas Search, Stream Processing, Migration, Security, Quickstart, Backup, Database Access, Network Access, Advanced, and New On Atlas (with 4 items). The main area displays a cluster named '707cluster'. It shows 'Database Deployments' with a 'CONNECT' button and a 'Data Size: 516.52 MB'. Below this are sections for 'Application Development' (with a note to 'OPTIMIZE YOUR CONNECTION POOL') and 'PyMongo v4.6.2'. A 'Toolbar' on the right provides links to various resources and documentation. At the bottom, there's a search bar and a 'New On Atlas' section.

- Create a cluster - click database under deployment, then click Build a Database

The screenshot shows the MongoDB Atlas Database Deployments page. The sidebar is identical to the previous screenshot. The main area is titled 'Database Deployments' and features a large 'Create a database' button with the sub-instruction 'Choose your cloud provider, region, and specs.' Below it is a note: 'Once your database is up and running, live migrate an existing MongoDB database into Atlas with our Live Migration Service.' At the bottom, there's a footer with links to Status, Terms, Privacy, Atlas Blog, and Contact Sales.

- Create cluster: cluster name created – 707Assessment



- Create user ID

Atlas Mira's Org Access Manager Billing

707 Assessment Data Services App Services Charts

Overview Deployment Services Security Quickstart

MIRA'S ORG - 2024-03-11 | 707 ASSESSMENT 2

Security Quickstart

To access data stored in Atlas, you'll need to create users and set up network security controls. Learn more about security setup.

How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.

Username and Password Certificate

Create User

Username: 707User Authentication Type: 707User Password: torriritmira

Your cluster has finished provisioning.

Where would you like to connect from?

Where would you like to connect from?

Enable access for any network(s) that need to read and write data to your cluster.

My Local Environment
Use this to add network IP addresses to the IP Access List. This can be modified at any time.

Cloud Environment
Use this to configure network access. Add to your cloud or on-premises environment. Specifically, set up IP Access Lists, Network Peering, and Private Endpoints.

Set your network security with any of the following options

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the Network Access Page.

IP Address
Description
Enter IP Address
Enter description
Add My Current IP Address
Add Entry

This IP address has already been added.

IP Access List
Description
128.236.195.204/32
EDIT REMOVE

VPC Peering
Peer your VPC with your Atlas cluster's VPC to ensure that traffic does not traverse the public internet. Requires an M0 cluster or higher.

Private Endpoint
Use your Private Endpoint to create a one-way connection from your VPC to your MongoDB Atlas database, ensuring Atlas cannot initiate connections back to your network. Requires a serverless instance, or an M0 cluster or higher.

Your cluster has finished provisioning.

Atlas - Miras Org - Overview - Deployment - Services - Security - Quickstart - Database - Data Lake - Device Sync - Triggers - Data API - Data Federation - Atlas Search - Stream Processing - Migration - New On Atlas - Goto

Access Manager - Billing

Overview - Deployment - Services - Security - Quickstart - Database - Data Lake - Device Sync - Triggers - Data API - Data Federation - Atlas Search - Stream Processing - Migration - New On Atlas - Goto

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the Network Access Page.

IP Address
Description
Enter IP Address
Enter description
Add My Current IP Address
Add Entry

This IP address has already been added.

IP Access List
Description
128.236.195.204/32

VPC Peering
Peer your VPC with your Atlas cluster's VPC to ensure that traffic does not traverse the public internet. Requires an M0 cluster or higher.

Private Endpoint
Use your Private Endpoint to create a one-way connection from your VPC to your MongoDB Atlas database, ensuring Atlas cannot initiate connections back to your network. Requires a serverless instance, or an M0 cluster or higher.

Configurable In New Tab Configurable In New Tab

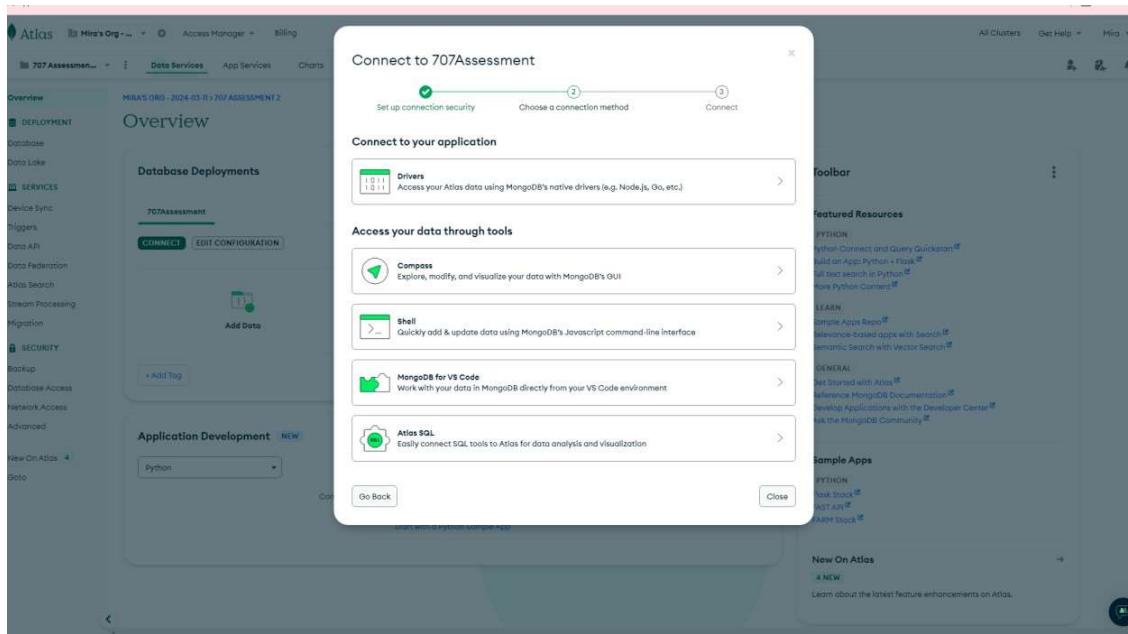
Congratulations on setting up access rules!
You will now be able to connect to your deployments. You can continue to add and update access rules in Database Access and Network Access.
 Hide Quickstart guide in the navigation. You can visit Project Settings to access it in the future.

Go to Overview

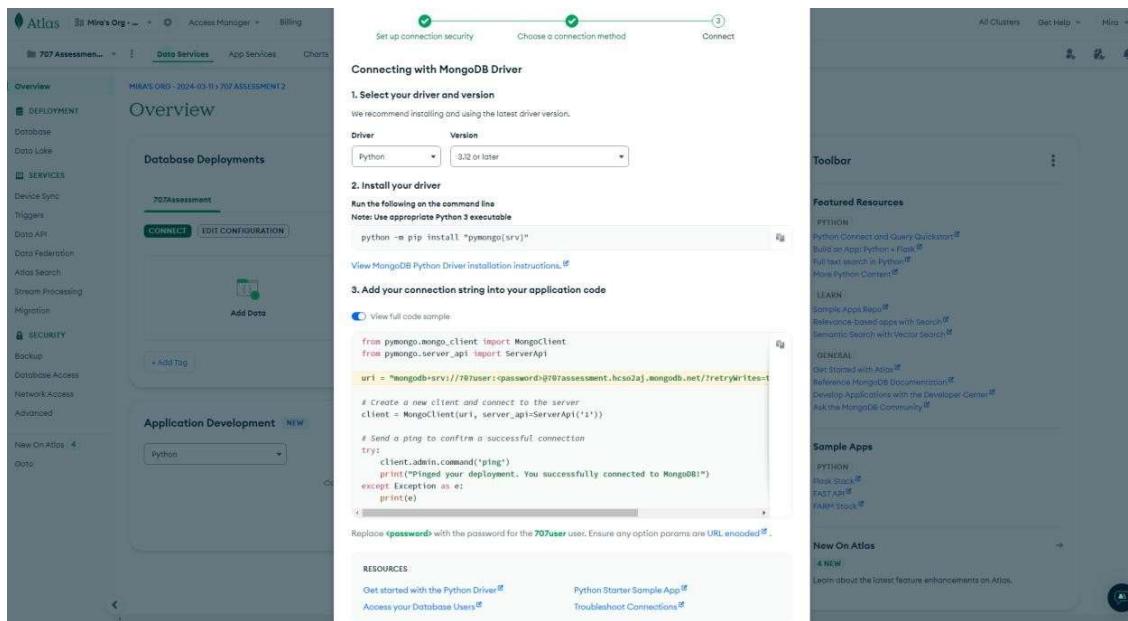
Finish and Close

Your cluster has finished provisioning.

- Setting up a connection to the database. Click drivers.



- Make sure the MongoDB is set to Python Ver. 3.12 later.



- Click Load Sample Data to browse the database collection.

The screenshot shows the MongoDB Atlas interface. In the top navigation bar, 'Atlas' is selected. Below it, '707 Assessment' is shown under 'All Clusters'. The main area is titled 'Overview' and displays 'Database Deployments' for the '707Assessment' cluster. It includes sections for 'CONNECT' and 'EDIT CONFIGURATION', and buttons for 'Add Data', 'Load Sample Data', and 'Data Modeling Templates'. On the right side, there's a 'Toolbar' with 'Featured Resources' (Python, Learn, General), 'Sample Apps' (Python, Flask Stock, FAST API, FARM Stock), and a 'New On Atlas' section.

- Once deployed, click Browse Collection.

This screenshot is identical to the previous one, showing the MongoDB Atlas Overview page. The '707 Assessment' cluster is selected. The 'Database Deployments' section now shows a green checkmark icon next to 'View deployment', indicating successful deployment. A tooltip message 'Sample dataset is deployed' is displayed. The rest of the interface, including the 'Toolbar' and 'New On Atlas' section, remains the same.

- Displaying sample MongoDB generated collection.

The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with various options like Overview, Database, Services, Security, and Data Lake. The main area is titled '707Assessment' and shows a database named 'sample_airbnb'. Underneath it, there's a collection named 'listingsAndReviews'. The interface displays storage details (Storage Size: 64.99MB, Logical Data Size: 88.99MB, Total Documents: 5858, Indexes Total Size: 65KB), and tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. A search bar at the top says 'Type a query: { field: 'value' }'. Below the tabs, there's a 'Filter' section and a 'QUERY RESULTS 1-20 OF MANY' section containing a single document. The document ID is '_id: "10006546"', and the name is 'Fibra Charming Duplex'. The document contains detailed information about a listing, including its location, amenities, and reviews.

```

{
  "_id": "10006546",
  "listing_url": "https://www.airbnb.com/rooms/10006546",
  "name": "Fibra Charming Duplex",
  "summary": "Fantastic duplex apartment with three bedrooms, located in the historical space. Privileged views of the Douro River and Ribeira square, our apartment is described as a unique place where you can feel the atmosphere of the neighborhood.",
  "neighborhood_overview": "In the neighborhood of the river, you can find several restaurants as well as \"lose yourself\" in the narrow streets and staircases zone, have lunch in \"transit\" (transport) + Metro station and S. Bento railway train + Bus stop a 50 m away from the house. We are always available to help guests, the house is fully available 24h/24h interaction: \"Cot - 10 € / night Dog - € 7,5 / night\" house_rules: \"Make the house your home...\" project_type: \"Entire home/apt\" room_type: \"Entire home/apt\" bed_type: \"Real Bed\" minimum_nights: \"27\" maximum_nights: \"27\" cancellation_policy: \"moderate\" last_scraped: 2019-02-16T05:00:00.000+00:00 calendar_last_scraped: 2019-02-16T05:00:00.000+00:00 first_review: 2010-01-01T05:00:00.000+00:00 last_review: 2019-01-28T05:00:00.000+00:00 accommodates: 3"
}

```

- Create a database and collection. Click + Create Database.

This screenshot is identical to the one above, showing the '707Assessment' database and the 'sample_airbnb.listingsAndReviews' collection in the MongoDB Atlas interface. It displays the same storage details, tabs, and query results for the 'Fibra Charming Duplex' listing.

- Name the database and collection. Click Create.

The screenshot shows the MongoDB Atlas interface. On the left sidebar, under 'Database', there is a 'Create Database' button. A modal window titled 'Create Database' is open, showing the database name 'Assessment2_707_Data_Engineering' and the collection name 'Amazon_Sales_Data'. The 'Create' button is visible at the bottom right of the modal. In the background, the main interface shows a list of databases and collections, including 'sample_airbnb' and 'sample_mflix'.

- Set up a connection from Python to MongoDB Atlas.

```
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi

def get_mongodbclient():
    userpassword = 'qxZAHkXprBqiAUy'
    uri = 'mongodb+srv://707user:u4NG9f4sCtn2mPhD@707assessment.hcso2aj.mongodb.net/?retryWrites=true&w=majority&appName=707Asses'

    # Create a new client and connect to the server
    client = MongoClient(uri, server_api=ServerApi('1'))

    # Send a ping to confirm a successful connection
    try:
        client.admin.command('ping')
        print("Pinged your deployment. You successfully connected to MongoDB!")
        return client
    except Exception as e:
        print(e)
        return None

    # Connect to MongoDB Atlas
    client = get_mongodbclient()
    # Initialize an instance of the database
    db = client['Assessment2_707_Data_Engineering']

Pinged your deployment. You successfully connected to MongoDB!
```

- Extracting the data from Python to MongoDB

```
# Load your dataset (e.g., from a CSV file)
# Source: https://www.kaggle.com/datasets/thedevastator/unlock-profits-with-e-commerce-sales-data
data = pd.read_csv('AmazonDF.csv')
# Due to limitation in Mongodb Atlas, I Limit the data to 100,000 rows
# data = data.head(200000)

# Access amazon_sales_report collection
collection = db['Amazon_Sales_Data']

# Reset the index if needed
data.reset_index(inplace=True)

# Set batch size due to timeout issue
batch_size = 10000

# Insert batches into MongoDB
for i in range(0, len(data), batch_size):
    batch = data.iloc[i:i + batch_size].to_dict('records')
    collection.insert_many(batch)

print(f"Inserted {len(data)} records into MongoDB Atlas.")
```

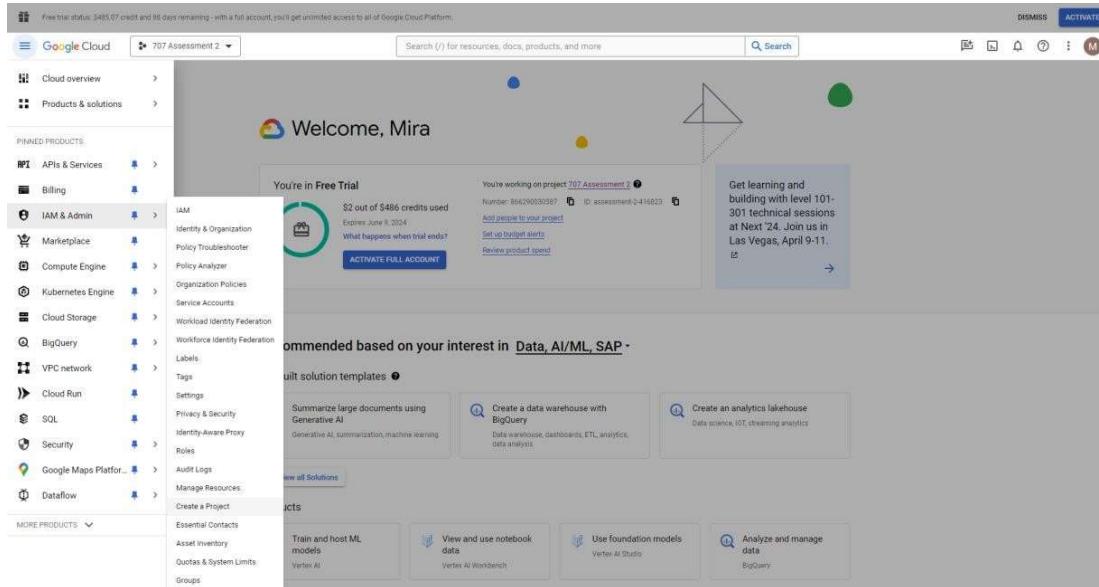
Inserted 128975 records into MongoDB Atlas.

- Checking in MongoDB. Go to the Collections tab.

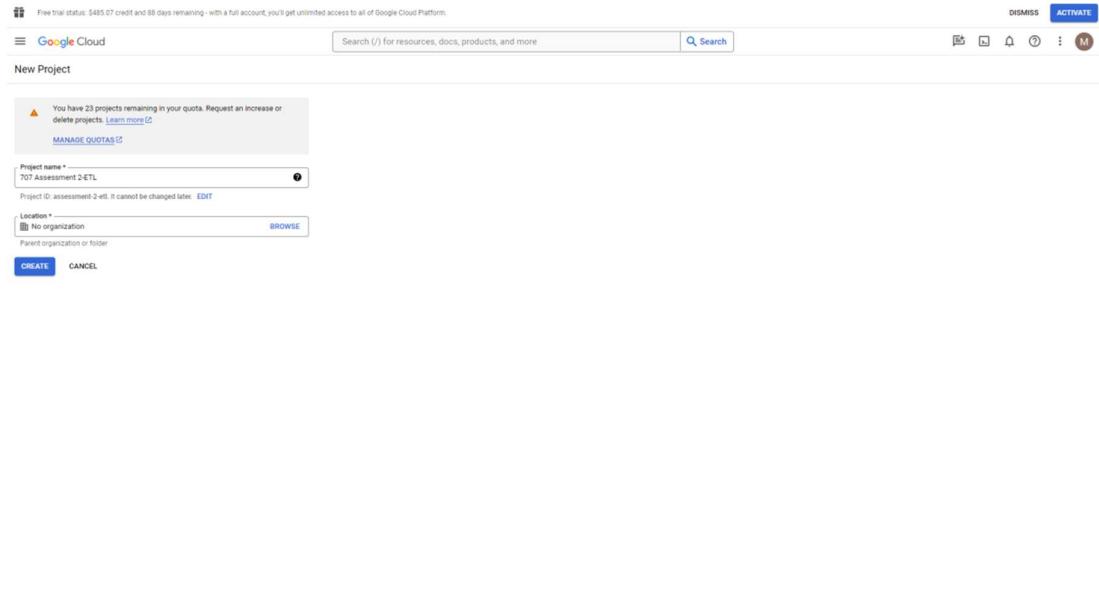
The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with 'DEPLOYMENT' selected under 'Database'. The main area shows '707Assessment' with 'OVERVIEW' and 'COLLECTIONS' tabs. Under 'COLLECTIONS', it lists 'Databases: 10' and 'Collections: 24'. One collection is highlighted: 'Assessment2_707_Data_E...', which has a preview pane showing a document with fields like '_id', 'level', 'order_id', 'product_id', 'order', 'Date', 'Status', 'Sales_Channel', 'Ship_Service_Level', 'Style', 'SKU', 'Category', 'Size', 'ASIN', 'Order_Created_At', 'Courier_Status', 'Qty', 'current_qty', 'Amount', 'ship-city', 'ship-state', 'ship-postal-code', and 'ship-country'. The interface also includes tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'.

MongoDB Atlas to BigQuery

- Create a new project in GCP – in the navigation menu, go to IAM & Admin and click Create a Project.



- Fill out the Project name and Location and click CREATE



A notification will pop out and will be seen on the right side of the page. Click SELECT PROJECT.

The screenshot shows the Google Cloud Platform dashboard for project "707 Assessment 2-ETL". The left sidebar lists various services like Cloud Storage, BigQuery, and Dataflow. The main area shows "Project info" with the project name "707 Assessment 2" and a "Virtual Machines" section. On the right, there's a "NOTIFICATIONS" sidebar with a timeline of recent events:

- Create Project: 707 Assessment 2-ETL (Just now)
- Deleting 3 object(s) and 0 folder(s) (16 minutes ago)
- Deleting 1 object(s) and 0 folder(s) (16 minutes ago)
- Deleting 0 object(s) and 1 folder(s) (26 minutes ago)
- Deleting 7 object(s) and 23 folder(s) (Failed to delete 1 folder(s)) (26 minutes ago)
- RETRY
- Deleting 1 object(s) and 5 folder(s) (26 minutes ago)
- Enable service: firestore.googleapis.com (707 Assessment 2) (3 hours ago)
- Enable service: cloudaicompassion.googleapis.com (707 Assessment 2) (12 hours ago)
- Enable service: bigquerydatatransfer.googleapis.com (707 Assessment 2) (2 days ago)
- Enable service: pubsub.googleapis.com (707 Assessment 2) (2 days ago)
- Enable service: cloudscheduler.googleapis.com (707 Assessment 2) (3 days ago)
- Enable service: datapipelines.googleapis.com (707 Assessment 2) (3 days ago)
- Enable service: dataflow.googleapis.com (707 Assessment 2) (3 days ago)

- Create a custom role – Go to IAM & Admin and click Roles

The screenshot shows the "PERMISSIONS" tab in the IAM & Admin section of the Google Cloud console. It displays the "Identity & Organization" section with a table of roles:

Name	Role	Security insights
grm.com	Mira	Owner

A message at the bottom states: "Now viewing project "707 Assessment 2-ETL" in organization "No organization"".

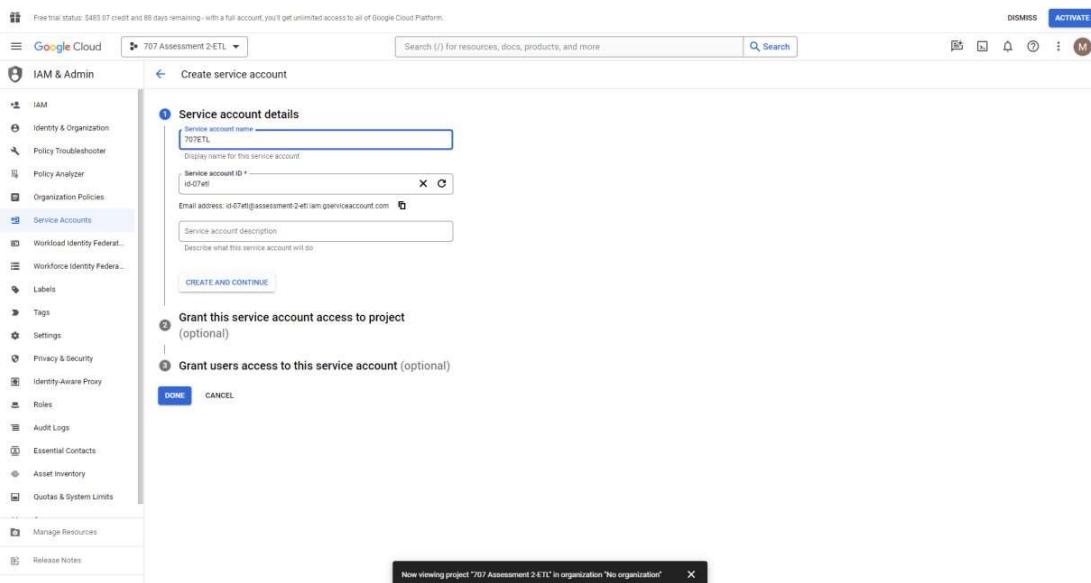
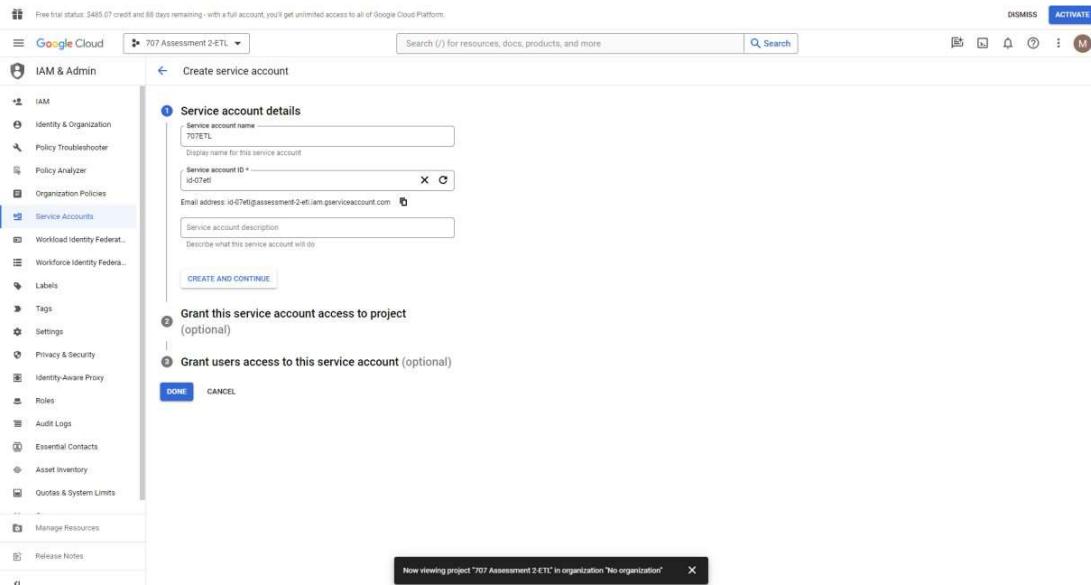
- Click + CREATE ROLE

The screenshot shows the Google Cloud IAM & Admin Roles page. The left sidebar is collapsed, and the main area displays a table of roles for the project "707 Assessment 2-ETL". The table columns are Type, Title, Used in, and Status. The roles listed include various Google services like Access Approval Approver, Access Approval Config Editor, Access Approval Invitator, Access Approval Viewer, Access Context Manager Admin, Access Context Manager Editor, Access Context Manager Reader, Access Transparency Admin, Actions Admin, Actions Viewer, Activity Analysis Viewer, Admin, Admins of Tenancy Units, Advisory Notifications Admin, Advisory Notifications Viewer, AI Platform Admin, AI Platform Developer, AI Platform Job Owner, AI Platform Model Owner, AI Platform Model User, AI Platform Notebooks Service Agent, AI Platform Operation Owner, and AI Platform Service Agent. Most roles are enabled.

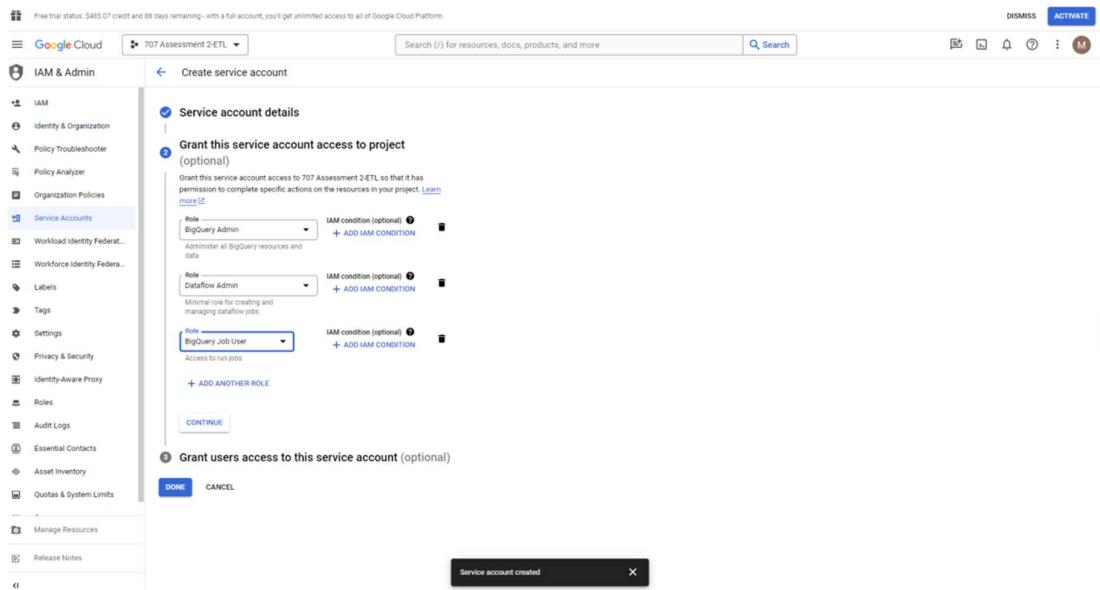
- Create a service account for the project. Go to IAM & Admin and click on Service Accounts.

The screenshot shows the Google Cloud IAM & Admin Permissions page. The left sidebar is collapsed, and the main area shows the permissions for the project "707 Assessment 2-ETL". It includes sections for NEW BY ROLES and GIVE ACCESS. A table lists service accounts with their names, roles, and security insights. The table includes rows for "Workload Identity Federation" and "Workforce Identity Federation". A checkbox for "Include Google-provided role grants" is present. A note at the top states: "Beginning on April 29th, 2024 all-scale policy analysis and advanced IAM recommendation capabilities will require Security Command Center Premium. Learn more" with a dismiss button.

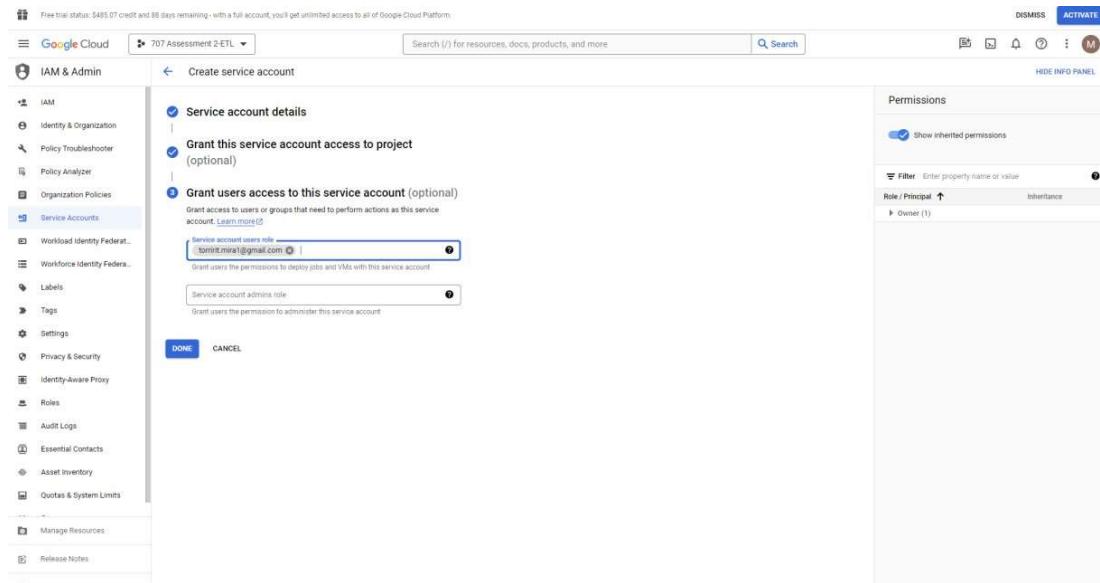
- Click **+CREATE SERVICE ACCOUNT**; provide the Service account name and click **CREATE AND CONTINUE**.



- Select three roles and click **CONTINUE**



- Provide the email address of the user and click **DONE**.



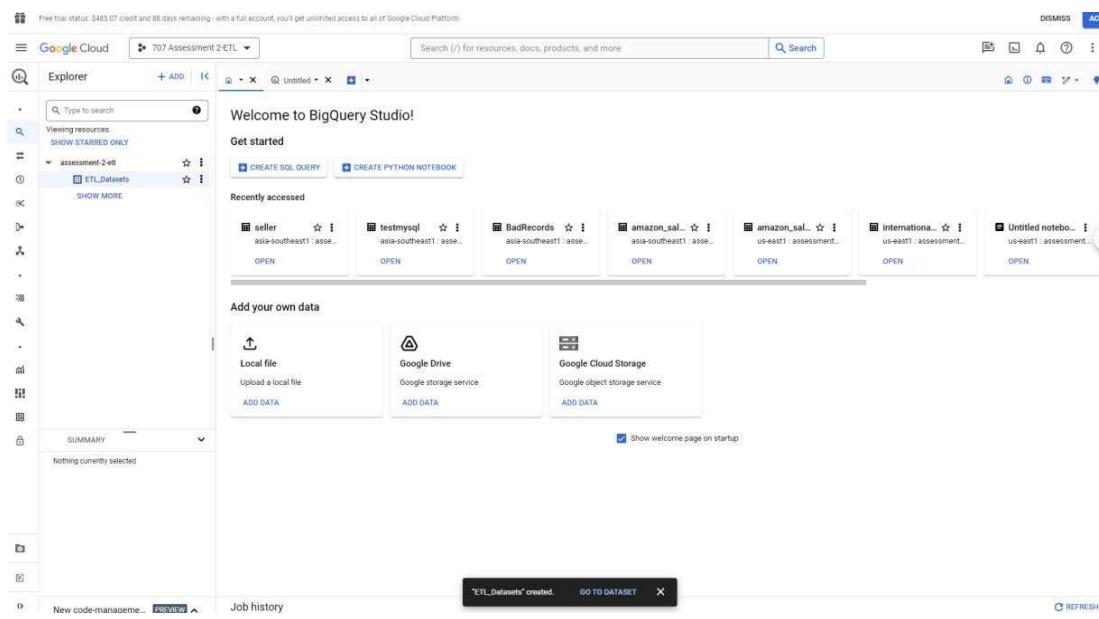
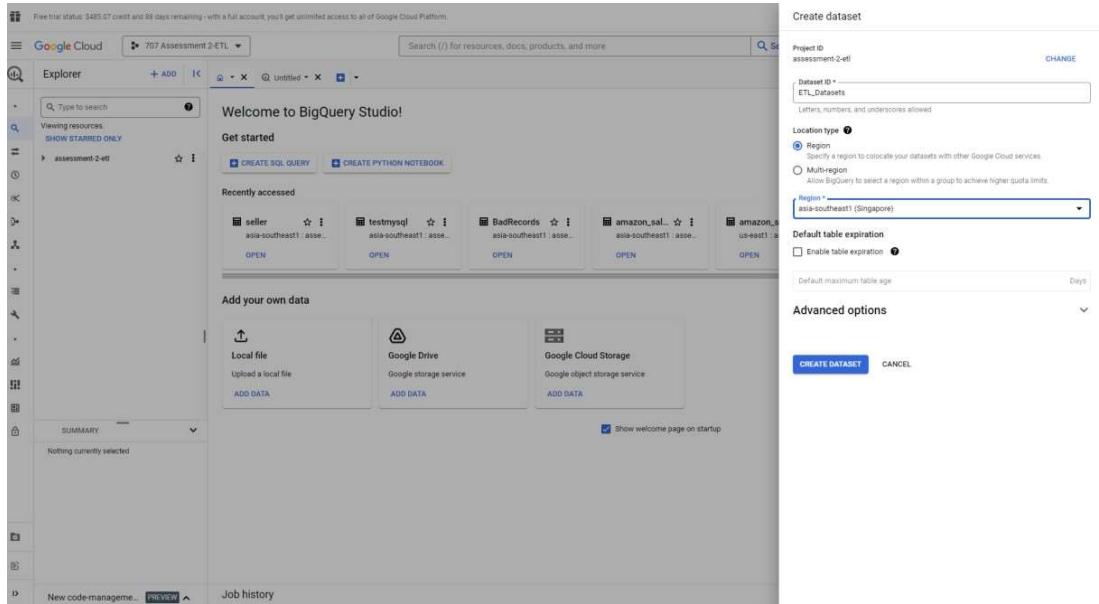
- Once the service account is created, go to IAM & Admin and click IAM to verify the new user.

The screenshot shows the Google Cloud IAM & Admin interface. The left sidebar is expanded to show various services like IAM, Identity & Organization, Policy Troubleshooter, etc. The main area is titled 'Permissions' for project '707 Assessment 2-ETL'. It lists several entries under 'VIEW BY PRINCIPALS'. One entry is for a service account: 'id-07e76f@assessment-2-ef1.iam.gserviceaccount.com' with roles 'BigQuery Admin', 'BigQuery Job User', and 'Dataflow Admin'. Another entry is for a user: 'torririt.mira1@gmail.com' with role 'Owner'. At the bottom right of the main area, there is a modal window with the message 'Policy updated'.

- Create a BigQuery Table – go to BigQuery and click BigQuery Studio

The screenshot shows the Google Cloud BigQuery interface. The left sidebar is expanded to show various products like Cloud overview, Products & solutions, APIs & Services, IAM & Admin, Marketplace, Compute Engine, Kubernetes Engine, Cloud Storage, and BigQuery. The 'BigQuery' section is selected. The main area is titled 'ANALYTICS' and shows a 'RECOMMENDATIONS HISTORY' section. Below it is a 'VIEW BY ROLES' section for project '707 Assessment 2-ETL'. It lists the same service account and user entries as the previous screenshot. At the bottom right of the main area, there is a modal window with the message 'Policy updated'.

- Click the database assessment-2-etl and create dataset. Create a Dataset ID. Choose the location type and Region. I chose the Southeast (Singapore) region as this is the region I put in my MongoDB then click **CREATE DATASET**.



- From your created dataset table, click the three dots beside it and click Create table. Fill out the needed data (with reference to the MongoDB collection).

The screenshot shows the Google BigQuery Studio interface. In the top navigation bar, there is a message about a free trial status: '\$485.07 credit and 88 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.' Below the navigation bar, the main area displays a 'Welcome to BigQuery Studio!' message and a 'Get started' section. On the left, the 'Explorer' sidebar shows a dataset named 'assessment-2-eft' with a sub-dataset 'ETL_Datasets' containing a table named 'amazon-sales-report'. A context menu is open over this table, with the 'Create table' option selected. Other options in the menu include 'Open in', 'Share', 'Copy ID', 'Refresh contents', 'Delete', and 'Local file'. The 'Local file' section contains options to 'Upload a local file' or 'ADD DATA'. To the right of the table list, there are sections for 'Google Drive' and 'Google Cloud Storage'. At the bottom of the screen, a notification bar says 'ETL_Datasets created. GO TO DATASET'.

- Provide the needed table name (should match with the table names in MongoDB Atlas)

The screenshot shows the Google BigQuery Studio interface focusing on the 'amazon-sales-report' table schema. The top navigation bar includes a message about a free trial status: '\$485.07 credit and 88 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.' The main area displays the table schema with various fields listed: index, Order_ID, Status, Fulfillment, Sales_Channel, ship_service_level, Style, SKU, Category, Size, ASIN, Courier_Status, Qty, currency, Amount, ship_city, ship_state, ship_postal_code, ship_country, promotion_ids, and B2B. Each field has its type (e.g., INTEGER, STRING, FLOAT), mode (e.g., NULLABLE, REQUIRED), and other properties like 'Key', 'Collation', 'Default Value', and 'Policy Tags'. Below the schema table, there are buttons for 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'. The left sidebar shows the dataset structure and a summary of the table's details, including last modified date (Mar 14, 2024) and location (asia-southeast1). The bottom navigation bar includes a 'REFRESH' button.

MongoDB Atlas Table (for reference)

The screenshot shows the MongoDB Atlas interface for the 'Amazon_Sales_Data' table. The left sidebar includes sections like Overview, Database, Services, Triggers, Data API, Security, and more. The main area displays the table schema and a sample document:

```

1 _id: ObjectId("65f894652a0bad4cb11b65")
2 Level_L0: 0
3 Unwound: 0
4 Indexes: 0
5 Score: -1.482-8678764-5731545j
6 Date: "2023-08-28T22:57:51Z"
7 Status: "Cancelled"
8 Fulfillment: "Merchant"
9 Sales_Channel: "Amazon"
10 ship_service_level: "Standard"
11 Style: "SET389"
12 Size: "L"
13 Category: "Set"
14 Size: "S"
15 ASIN: "B09XK9D7Z"
16 Courier_Status: "Cancelled"
17 Qty: 0
18 currency: "INR"
19 amount: 647.62
20 ship_city: "MUMBAI"
21 ship_state: "MAHARASHTRA"
22 ship_postal_code: 400093
23 ship_country: "IN"

```

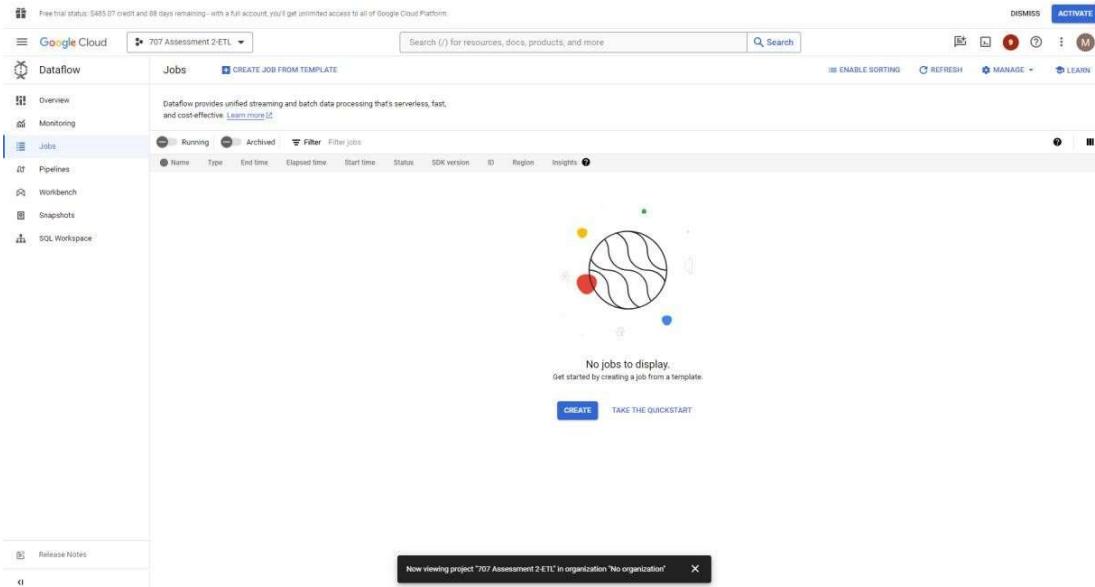
Below the table, there's a query results section showing 1-20 of many results.

- Once the table is created, you may create a Dataflow job. Go to Dataflow and click on Jobs.

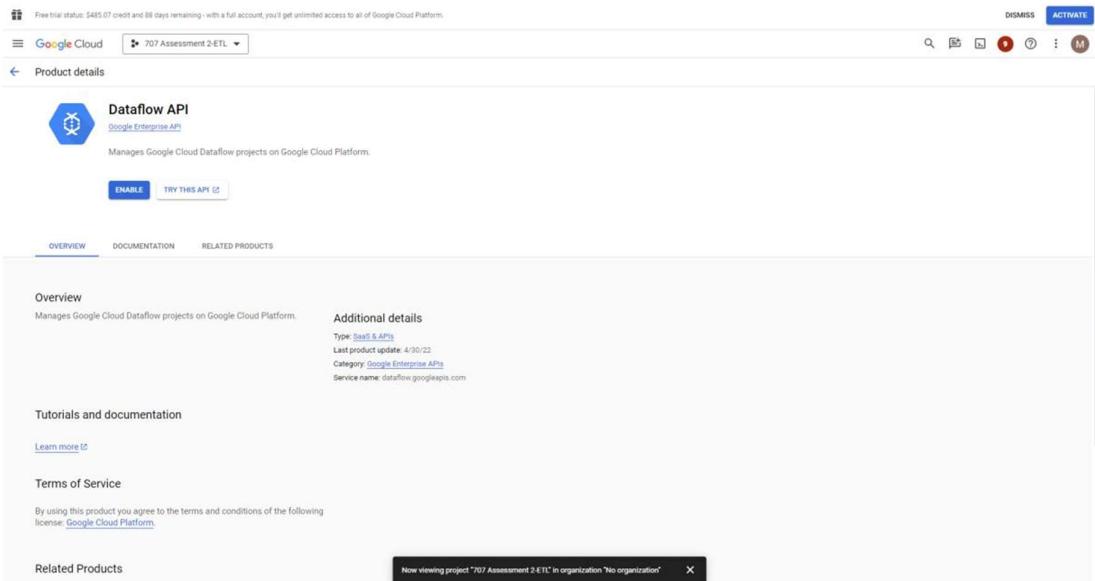
The screenshot shows the Google Cloud Dataflow interface for the 'amazon-sales-report' table. The left sidebar lists various Google Cloud services like APIs & Services, Billing, IAM & Admin, Marketplace, Compute Engine, Kubernetes Engine, Cloud Storage, BigQuery, VPC network, Cloud Run, SQL, Security, Google Maps Platform, and Dataflow. The Dataflow section is expanded, showing jobs and pipelines. The main area displays the table schema:

Field name	Type	Mode	Key	Collation	Default Value	Policy Tag	Description
_id	INTEGER	NULLABLE	-	-	-	-	-
Order_ID	STRING	NULLABLE	-	-	-	-	-
Status	STRING	NULLABLE	-	-	-	-	-
Fulfillment	STRING	NULLABLE	-	-	-	-	-
Sales_Channel	STRING	NULLABLE	-	-	-	-	-
ship_service_Level	STRING	NULLABLE	-	-	-	-	-
Style	STRING	NULLABLE	-	-	-	-	-
SKU	STRING	NULLABLE	-	-	-	-	-
Category	STRING	NULLABLE	-	-	-	-	-
Size	STRING	NULLABLE	-	-	-	-	-
ASIN	STRING	NULLABLE	-	-	-	-	-
Courier_Status	STRING	NULLABLE	-	-	-	-	-
Qty	INTEGER	NULLABLE	-	-	-	-	-
currency	STRING	NULLABLE	-	-	-	-	-
amount	FLOAT	NULLABLE	-	-	-	-	-
ship_city	STRING	NULLABLE	-	-	-	-	-
ship_state	STRING	NULLABLE	-	-	-	-	-
ship_postal_code	INTEGER	NULLABLE	-	-	-	-	-
ship_country	STRING	NULLABLE	-	-	-	-	-
promotion_ids	STRING	NULLABLE	-	-	-	-	-
B2B	BOOLEAN	NULLABLE	-	-	-	-	-

- Click +CREATE JOB FROM TEMPLATE



- Click **ENABLE** Dataflow API



- Create a job from a template

Job name * m707

Region endpoint * asia-southeast1 (Singapore)

Dataflow template * MongoDB to BigQuery

Required Parameters

MongoDB Connection URL * `mongodb://172.17.0.2:49152@172.17.0.2:49152:172.17.0.2:49152@172.17.0.2:49152/assessment2-707-assessment-eldgap.mongodb.net/`

MongoDB database * Assessment2_707_Data_Engineering

MongoDB collection * Amazon_Sales_Data

User option * PLATEN

Repository output table * assessment2-707_ETL_Datasets.amazon-sales-report

Encryption

Google-managed encryption key

Customer-managed encryption key (CMK)

Optional Parameters

RUN JOB

Job creation may take a while.

Job info

Job name etl707

Job ID 2024-03-13_16_47_54-839181753661402471

Job type -

Job status Downed

Job region asia-southeast1

Current workers -

Latest worker status -

Start time March 14, 2024 at 12:47:57 PM GMT+13

Elapsed time 1 min 53 sec

Encryption type Google-managed

Dataflow Prime -

Runner v2 -

Resource metrics

Current vCPUs -

Total vCPU time -

Current memory -

Total memory time -

Current HDD PO -

Total HDD PO time -

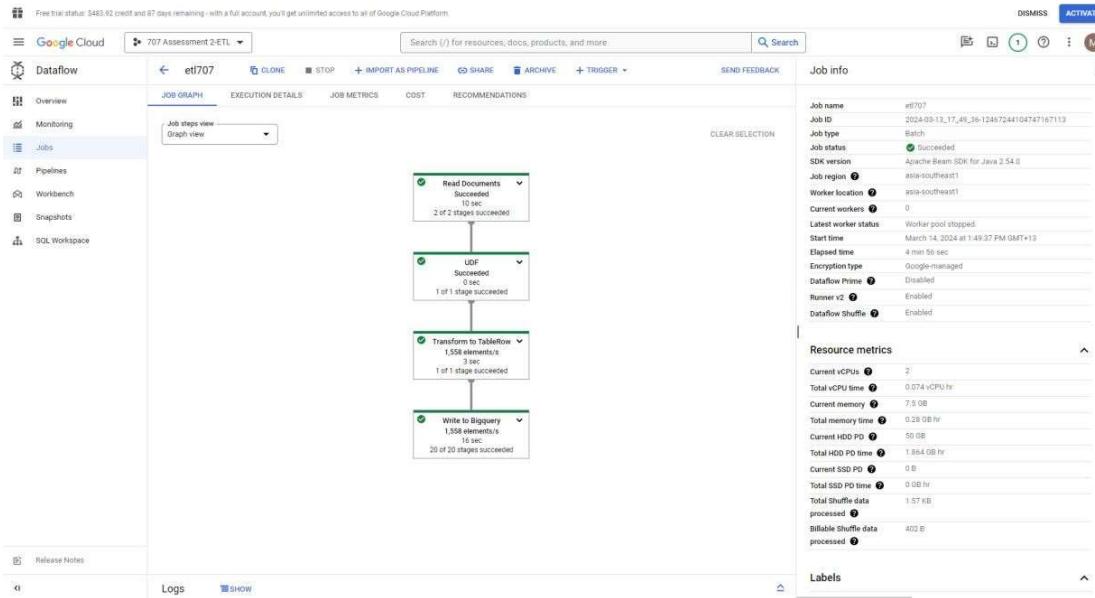
Current SSD PO -

Total SSD PO time -

Equivalent REST

Job creation may take a while...

- Successful ETL process.



The same process should be used for the second dataset, but this time, I used the GCP bucket as a data source instead of MongoDB Atlas.

Sales Transaction Dataset – From Python to GCP Bucket

- Import storage library in Python to import the cleaned dataset to GCP Bucket.

```
from google.cloud import storage

# Load service account credentials
credentials_path = 'assessment-2-etl-95c16a484157.json'
storage_client = storage.Client.from_service_account_json(credentials_path)

# Bucket name
bucket_name = 'assessment2_bucket'
# Get the bucket
bucket = storage_client.get_bucket(bucket_name)

# Specify the remote filename in the bucket
blob = bucket.blob('Cleaned_SalesTransactionDF.csv')

# Upload the local file
blob.upload_from_filename('./Cleaned_SalesTransactionDF.csv')
```

- Create a Bucket in GCP – in the navigation menu, go to Cloud Storage and click Buckets.

- On the upper left side, click +CREATE.

* Provide the needed details.

Free trial status: \$485.07 credit and 87 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

Create a bucket

Name your bucket
Name: assessment2_bucket

Choose where to store your data
This choice defines the geographic placement of your data and affects cost, performance, and availability. Cannot be changed later. [Learn more](#)

Location type
 Multi-region (highest availability across largest area)
 Dual-region (high availability and low latency across 2 regions)
 Region (lowest latency within a single region)
Selected: asia-southeast1 (Singapore)

Good to know

Location pricing
Storage rates vary depending on the storage class of your data and location of your bucket. [Pricing details](#)

Current configuration: Region / Standard

Item	Cost
asia-southeast1 (Singapore)	\$0.020 per GB-month

ESTIMATE YOUR MONTHLY COST

CONTINUE

Choose a storage class for your data
Default storage class: Standard

Choose how to control access to objects
Public access prevention: On
Access control: Uniform

Choose how to protect object data
Soft delete policy: Enabled
Object versioning: Disabled
Bucket retention policy: Disabled
Object retention: Disabled
Encryption type: Google-managed

CREATE **CANCEL**

Click **CONTINUE** and **CREATE**

Free trial status: \$485.07 credit and 87 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

Create a bucket

Default storage class: Standard

Choose how to control access to objects
Public access prevention: Off
Access control: Uniform

Choose how to protect object data
Your data is always protected with Cloud Storage but you can also choose from these additional data protection options to add extra layers of security.

Data protection

Soft delete policy (For data recovery)
When enabled, deleted objects will be kept for a specified period after they're deleted and can be restored during this time. [Learn more](#)

Object retention period
Duration: 90 days

Object versioning (For version control)
For preventing data loss or overwriting data. To minimize the cost of storing versions, we recommend limiting the number of noncurrent versions per object and scheduling them to expire after a number of days. [Learn more](#)

Max. number of versions per object
1

If you want overwrite protection, increase the count to at least 2 versions per object. Version count includes live and noncurrent versions.

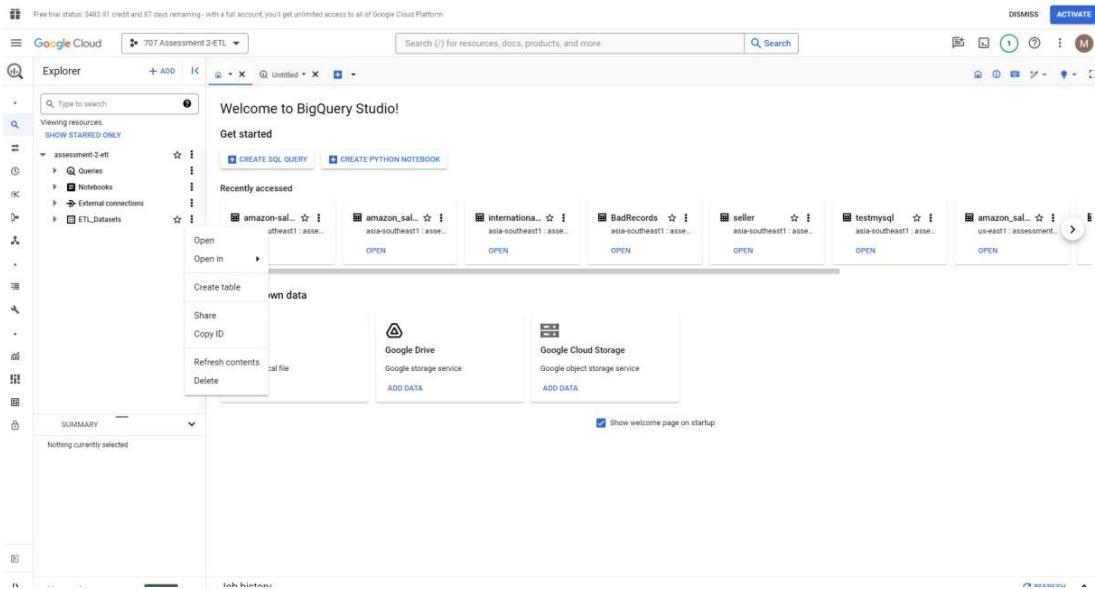
Expire noncurrent versions after
1 days
7 days recommended for Standard storage class

Retention (For compliance)
For preventing the deletion or modification of the bucket's objects for a specified period of time

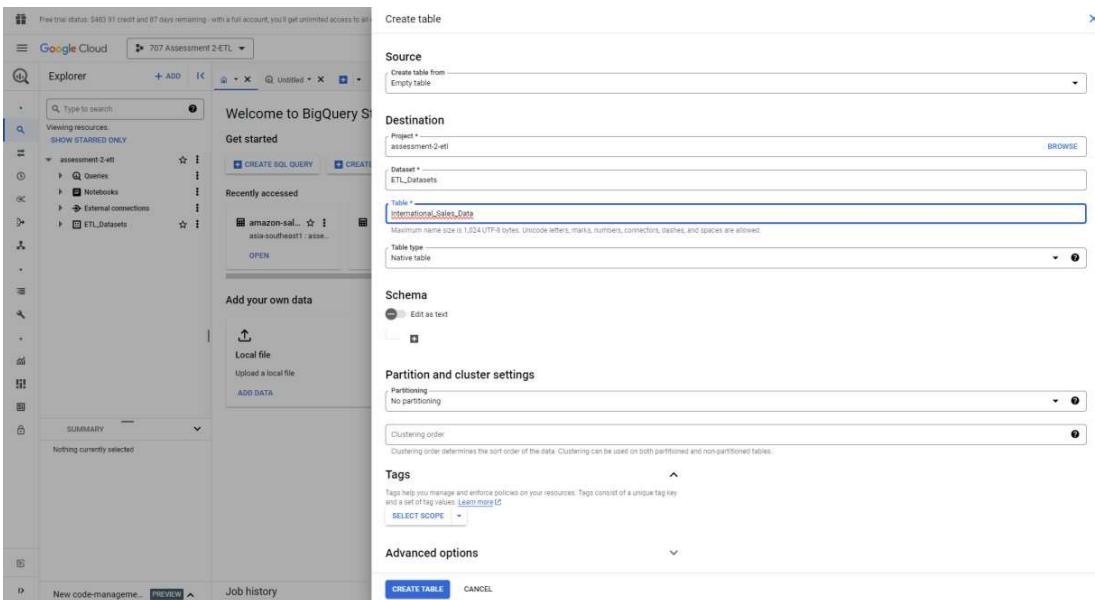
DATA ENCRYPTION

CREATE **CANCEL**

- After creating a bucket, create a table.



Naming the table



- Go to the table and **EDIT SCHEMA**

The screenshot shows the Google Cloud BigQuery schema editor for the 'International_Sales_Data' table. The table has the following schema:

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
TransactionNo	STRING	NULLABLE	-	-	-	-	-
Date	DATE	NULLABLE	-	-	-	-	-
ProductNo	INTEGER	NULLABLE	-	-	-	-	-
Price	FLOAT	NULLABLE	-	-	-	-	-
Quantity	INTEGER	NULLABLE	-	-	-	-	-
CustomerNo	STRING	NULLABLE	-	-	-	-	-
Country	STRING	NULLABLE	-	-	-	-	-
Timestamp	TIMESTAMP	NULLABLE	-	-	-	-	-

After creating the table in the BigQuery, we will now create a Job using the Dataflow template text file to BigQuery.

Go to Dataflow, CREATE JOB FROM TEMPLATE

The screenshot shows the Google Cloud Dataflow Jobs page. There are several running and failed jobs listed:

Name	Type	End time	Elapsed time	Start time	Status	SDK version	ID	Region	Insights
etl707	Batch	Mar 14, 2024, 1:54:31PM	4 min 56 sec	Mar 14, 2024, 1:49:37PM	Succeeded	2.54.0	2024-03-13_17_49_36-12467244104747167113	asia-southeast1	
etl707	Batch	Mar 14, 2024, 1:49:07PM	5 min 51 sec	Mar 14, 2024, 1:46:16PM	Failed	2.54.0	2024-03-13_17_43_16-7523463620109382234	asia-southeast1	
etl707	Batch	Mar 14, 2024, 1:42:47PM	5 min 58 sec	Mar 14, 2024, 1:39:49PM	Failed	2.54.0	2024-03-13_17_36_48-12198499833840394917	asia-southwest1	
etl707	Batch	Mar 14, 2024, 1:35:56PM	6 min 5 sec	Mar 14, 2024, 1:29:51PM	Failed	2.54.0	2024-03-13_17_29_51-118779875800608060607	asia-southwest1	
etl707	Batch	Mar 14, 2024, 1:29:38PM	6 min 7 sec	Mar 14, 2024, 1:23:31PM	Failed	2.54.0	2024-03-13_17_23_30-1796494252630791715	asia-southwest1	
etl707	Batch	Mar 14, 2024, 1:19:43PM	6 min 3 sec	Mar 14, 2024, 1:13:40PM	Failed	2.54.0	2024-03-13_17_13_39-11355460333358216768	asia-southeast1	
etl707	Batch	Mar 14, 2024, 1:13:53PM	5 min 55 sec	Mar 14, 2024, 1:07:18PM	Failed	2.54.0	2024-03-13_17_07_17-8370746714467926350	asia-southwest1	
etl707	Batch	Mar 14, 2024, 1:04:09PM	6 min 31 sec	Mar 14, 2024, 12:57:38PM	Failed	2.54.0	2024-03-13_16_57_37-1569716900440904147	asia-southwest1	

- Creating a job from a template.

Job name: International_Sales_Data

Regional endpoint: asia-southeast1 (Singapore)

Dataflow template: Text Files on Cloud Storage to BigQuery

Required Parameters

- Cloud Storage input file(s):
- Cloud Storage location of your BigQuery schema file, described as:
- BigQuery output table:
- Temporary directory for BigQuery loading process:
- Temporary location:

Additional information

The costs of this batch pipeline will depend on the amount of data you will process.

Release Notes

Now viewing project "707 Assessment 2-ETL" in organization "No organization"

Uploaded the cleaned dataset

assessment2_bucket

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history
Cleaned_SalesTransactions.csv	45.5 MB	application/vnd.ms-excel	Mar 14, 2024, 6:00:42 PM	Standard	Mar 14, 2024, 6:00:42 PM	Not public	-

Get started with Cloud Storage

- Getting bucket information
- Uploading objects
- Downloading objects
- Use cases for Cloud Storage
- Terraform samples
- Architecture guides for storage
- Prevent public access
- Making data public

- Creating a Job template for source file (first parameter)

- Mapping between CSV and BigQuery

```

{
  "name": "TransactionNo",
  "type": "STRING"
},
{
  "name": "Date",
  "type": "DATE"
},
{
  "name": "ProductName",
  "type": "STRING"
},
{
  "name": "Price",
  "type": "FLOAT"
},
{
  "name": "Quantity",
  "type": "INTEGER"
},
{
  "name": "CustomerNo",
  "type": "STRING"
}
]
}

```

- Upload the JSON file to the bucket

The screenshot shows the Google Cloud Storage interface. On the left, there's a sidebar with 'Buckets', 'Monitoring', and 'Settings'. The main area shows a bucket named 'assessment2_bucket'. It has a location of 'asia-southeast1 (Singapore)', a storage class of 'Standard', and 'Public access' set to 'Not public'. Below this, there's a table of objects. One object is listed: 'Cleaned_SalesTransactions.xlsx' (45.5 MB, application/vnd.ms-excel, created Mar 14, 2024). The right side of the screen has a sidebar titled 'Get started with Cloud Storage' with various links like 'Uploading objects', 'Downloading objects', 'Use cases for Cloud Storage', etc.

The screenshot shows a Windows File Explorer window. The path is 'This PC > Desktop > Mira > Data Analytics > 707 Data Engineering > Assessment 2'. In the center, there's a folder named 'Assessment 2' which contains several files: 'Dataset', 'big_query_schema.json', 'ETL from BigQuery', 'GODA707-Assessment 2', 'mapping script', 'Mix_report_assessment2', 'Python to Hive', 'Python to MongoDB Atlas', and 'Reason for not using Cassandra'. The 'big_query_schema.json' file is highlighted. The right side of the screen has a sidebar titled 'Get started with Cloud Storage' with various links like 'Uploading objects', 'Downloading objects', 'Use cases for Cloud Storage', etc.

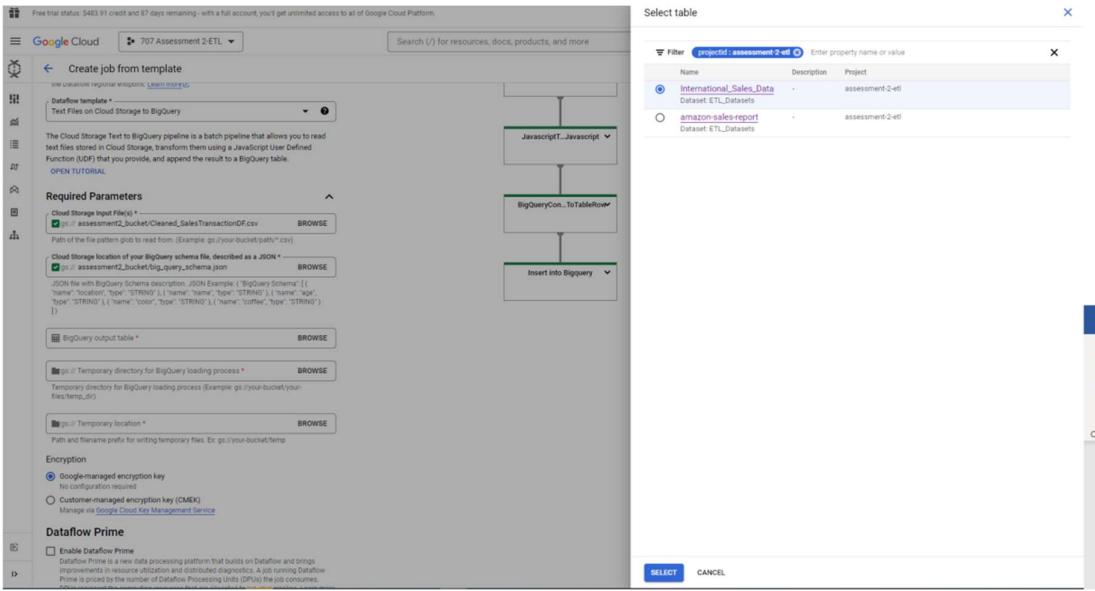
- Checking if it was successfully uploaded

The screenshot shows the Google Cloud Storage interface for the bucket 'assessment2_bucket'. It displays two files: 'Cleaned_SalesTransactionDF.csv' and 'big_query_schema.json'. A success message '1 file successfully uploaded' is visible at the bottom.

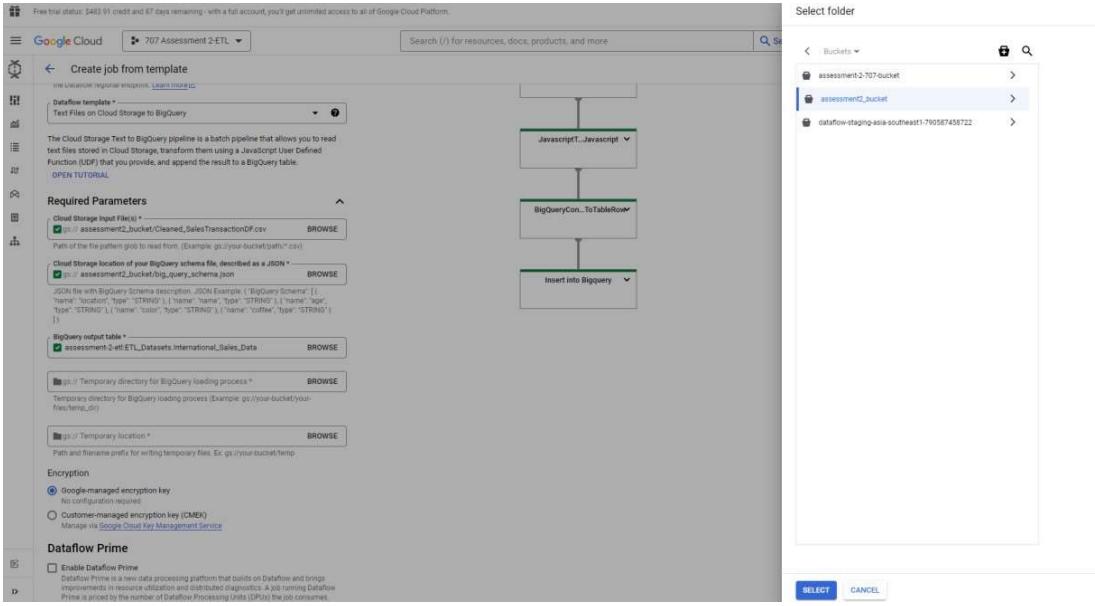
- Go back to Dataflow create job. Select the big_query_schema (second parameter)

The screenshot shows the 'Create job from template' page. In the 'Required Parameters' section, the 'Cloud Storage Input File(s)' field contains 'gs://assessment2_bucket/Cleaned_SalesTransactionDF.csv'. The 'big_query_schema.json' field is highlighted with a red border and has an error message 'Error: value is required'. The 'Additional information' section shows a pipeline diagram starting with 'Read from source' and ending with 'Insert into BigQuery'.

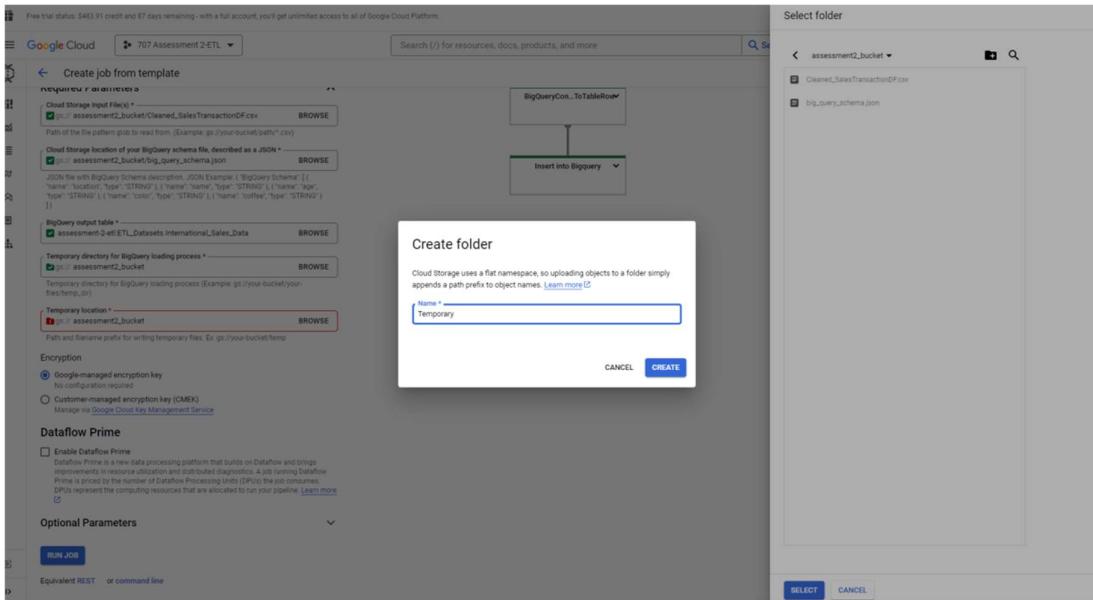
- Loading the transformed data from CSV to BigQuery. Select the dataset. (third parameter)



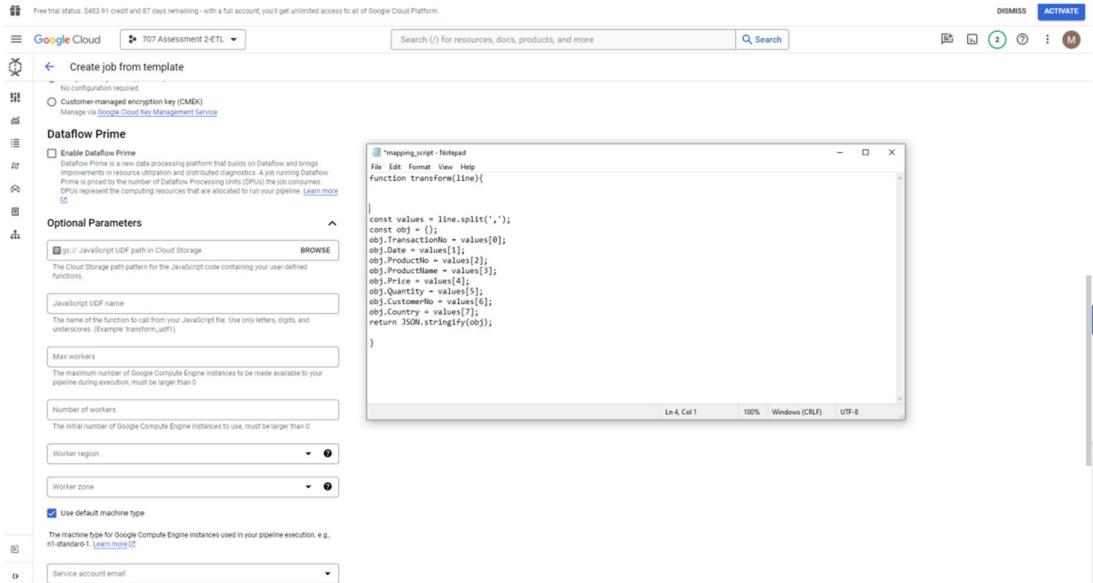
- Select the Bucket to be used as temporary location for loading process



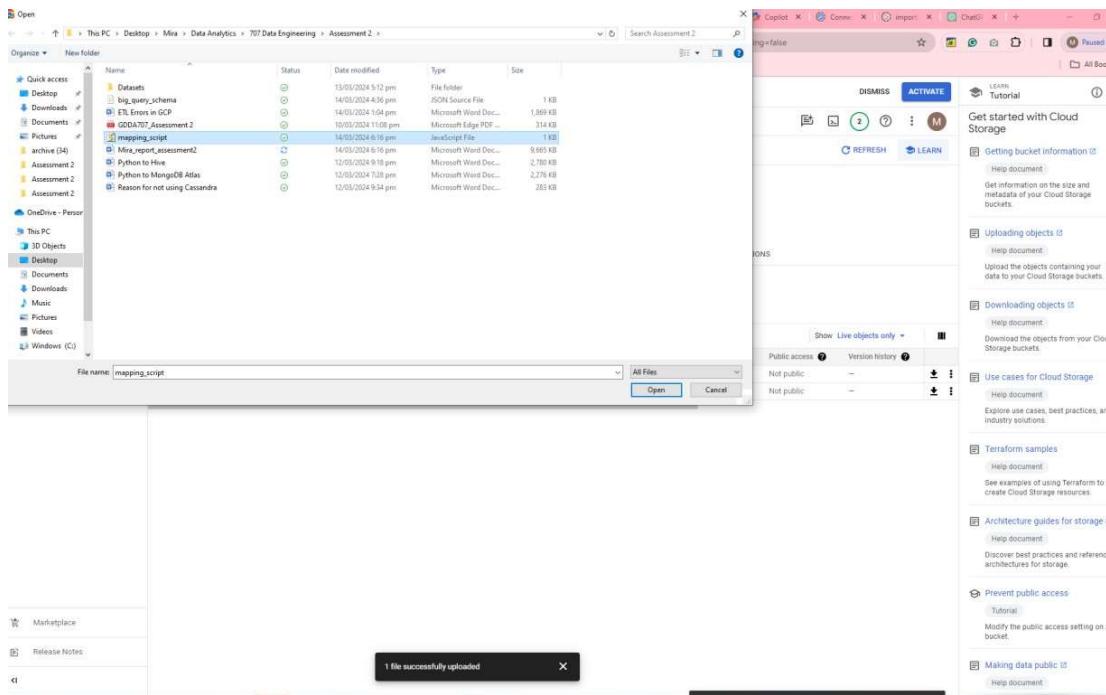
- Create temporary location for writing temporary files needed in the loading process



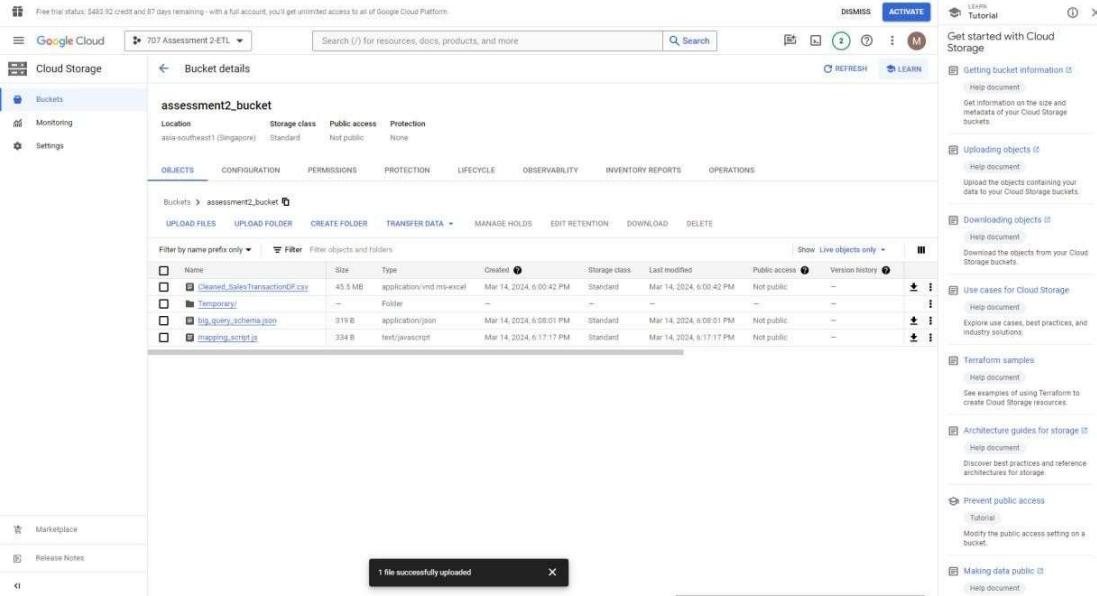
- Click Optional Parameters and go to JavaScript UDF name



- Go to the Bucket and upload the mapping script file



- Check if successfully uploaded



- Go back to the job creation page under Optional Parameters, browse and choose the mapping_script.js

The screenshot shows the 'Create job from template' interface in Google Cloud Dataflow Prime. In the 'Optional Parameters' section, the 'JavaScript UDF path in Cloud Storage' field is populated with 'gs://assessment2_bucket/mapping_script.js'. A modal window titled 'Select object' is overlaid, displaying the contents of the 'assessment2_bucket' storage location, including 'mapping_script.js' which is currently selected.

- Type ‘transform’ in the JavaScript UDF name and click **RUN JOB**

The screenshot shows the 'Create job from template' interface in Google Cloud Dataflow Prime. The 'Optional Parameters' section has been updated to show 'transform' in the 'JavaScript UDF name' field. The other parameters remain the same as the previous screenshot, including the Cloud Storage path for the UDF file.

- Running the job

Job info

Job name	csttobigquery
Job ID	2024-03-19_22_21_29-788346405677683690
Job type	Batch
Job status	Starting...
SDK version	Apache Beam SDK for Java 2.54.0
Job region	asia-southwest1
Worker location	asia-southwest1
Current workers	0
Latest worker status	March 14, 2024 at 6:21:31 PM GMT+13
Start time	9 sec
Elapsed time	Google-managed
Dataflow Prime	Disabled
Runner v2	Enabled
Dataflow Shuffle	Enabled

Resource metrics

Current vCPUs	-
Total vCPU time	- vCPU hr
Current memory	-
Total memory time	- 0B hr
Current HDD PD	-
Total HDD PD time	- 0B hr
Current SSD PD	-
Total SSD PD time	- 0B hr

Labels

goog-dataflow-provided-template-name	gcs_text_to_bigquery
goog-dataflow-named-template-time	legacy

- Successfully done ETL

Job info

Job name	csttobigquery
Job ID	2024-03-19_22_21_29-788346405677683694
Job type	Batch
Job status	Succeeded
SDK version	Apache Beam SDK for Java 2.54.0
Job region	asia-southwest1
Worker location	asia-southwest1
Current workers	0
Latest worker status	Worker pool stopped
Start time	March 14, 2024 at 7:52:49 PM GMT+13
Elapsed time	6 min 39 sec
Encryption type	Google-managed
Dataflow Prime	Disabled
Runner v2	Enabled
Dataflow Shuffle	Enabled

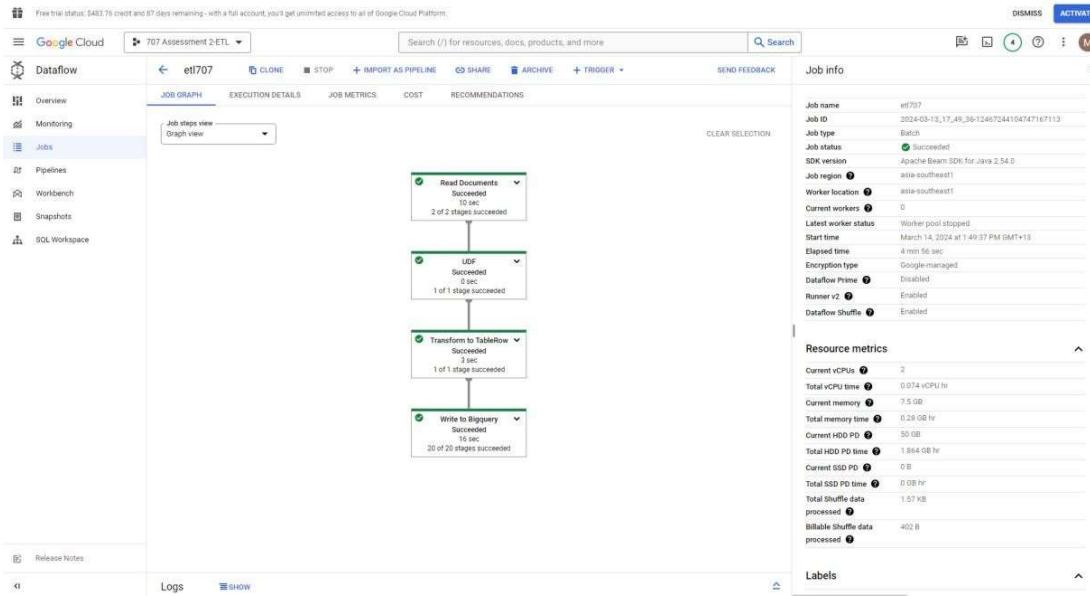
Resource metrics

Current vCPUs	1
Total vCPU time	0.098 vCPU hr
Current memory	3.75 GB
Total memory time	0.368 GB hr
Current HDD PD	25.0B
Total HDD PD time	2.454 GB hr
Current SSD PD	0B
Total SSD PD time	0 GB hr
Total shuffle data processed	3.88 kB
Billable shuffle data processed	987 B

Labels

Release Notes	
Logs	SHOW

- Successfully created ETL

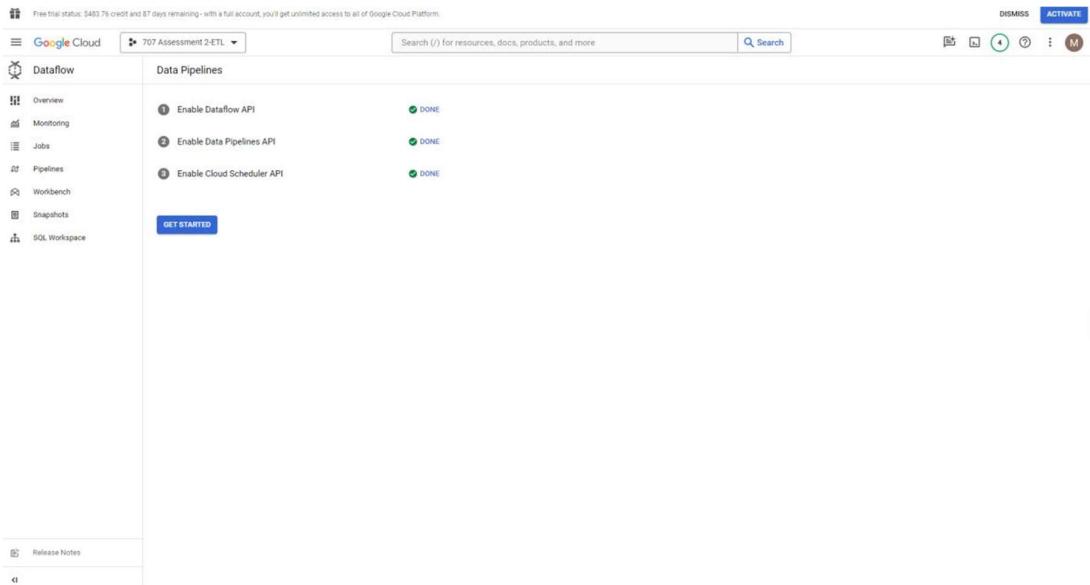


Creating a Pipeline to run a scheduled job

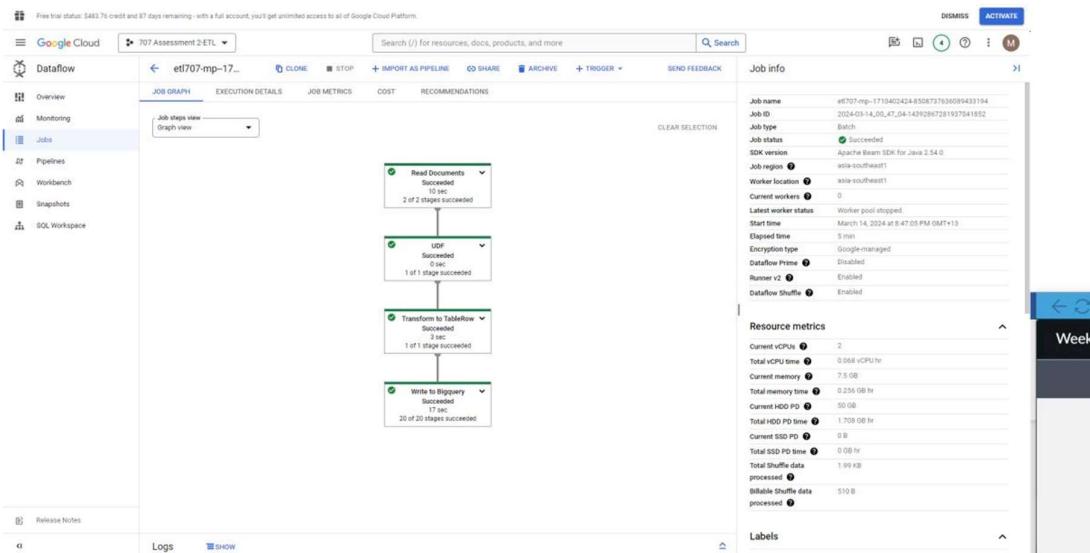
- On the same previous page, click **+IMPORT AS PIPELINE**, then click **CREATE PIPELINE**

The screenshot shows the 'Create pipeline from template' page. It includes fields for Pipeline name (etl707), Pipeline ID (etl707_EDIT), Regional endpoint (asia-southeast1 (Singapore)), and Dataflow template (MongoDB to BigQuery). The 'Additional information' section contains fields for scheduling the pipeline: Run every (Weekly), on day of week (Sunday), At time (12:00 am), and Timezone (New Zealand Daylight Time (NZDT)). There's also a note about running at 12:00 AM NZDT. At the bottom, there's a 'Required Parameters' section with a note about MongoDB connection URIs.

- Running the Pipeline



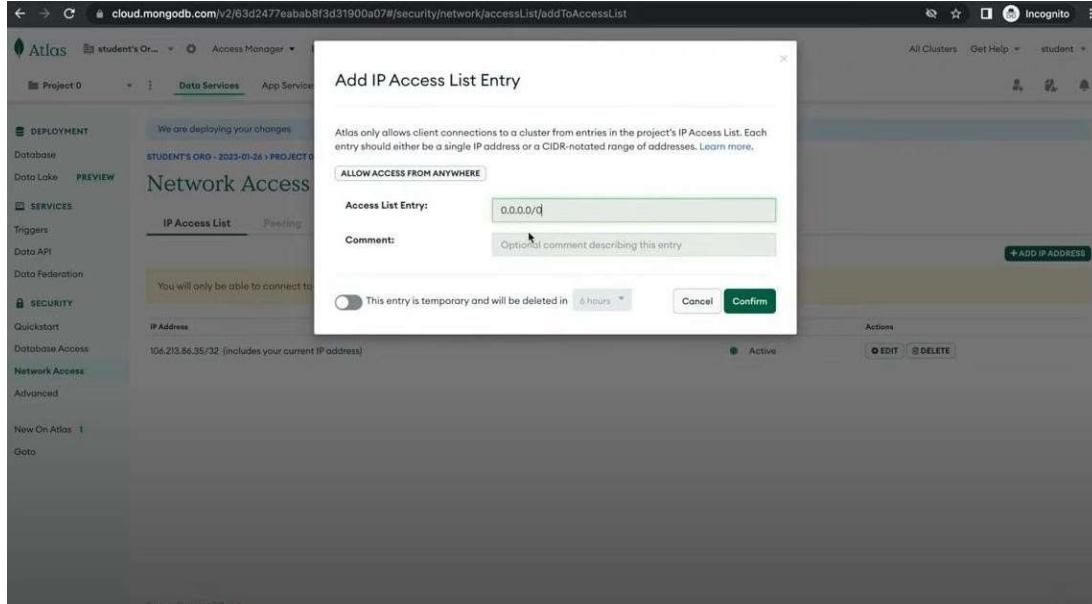
- Successfully created Pipeline



Challenges during ETL and Integration

Some of the challenges I encountered are the following:

1. MongoDB connection to GCP – I added the IP address 0.0.0.0/0 in the MongoDB Atlas to allow access from anywhere since I could not determine the GCP IP address.



The screenshot shows the 'IP Access List' section of the MongoDB Atlas Network Access page. It displays two entries: one for the current IP address (106.213.86.35/32) and another for 0.0.0.0/0 (marked as temporary). Both entries are listed with 'Active' status and edit/delete actions.

IP Address	Comment	Status	Actions
106.213.86.35/32	(includes your current IP address)	Active	EDIT DELETE
0.0.0.0/0	(includes your current IP address)	Active	EDIT DELETE

2. There are many errors in the Dataflow job creation in GCP due to schema, connections, and job settings. To solve these problems, I double-checked the field names in MongoDB and BigQuery and made sure they had the same schema. On the connections, I made sure that I provided the correct role, credentials (username and password), and the correct server address (connection strings). On the job settings, I made sure to choose the correct Dataflow templates and correct parameters.

Data Integration

Mira Torritit

JOB LOGS **WORKER LOGS** **DIAGNOSTICS** **DATA SAMPLING**

Severity: Error ▾ **Filter** Search all fields and values **MAX TIME**

SEVERITY	TIMESTAMP	SUMMARY
!!	2024-03-12 11:48:25.584 NZDT	Failed to read the result file : gs://dataflow-staging-asia-east1-866290030587/staging/template_launches/2024-03-11_15_45_52-3889315570494279297/operation_result with error message: (6600fcfb2f6cf71f): Unable to open template file: gs://dataflow-staging-asia-east1-866290030587/staging/template_launches/2024-03-11_15_45_52-3889315570494279297/operation_result..

Open in Logs Explorer

```
{
  "insertId": "bdoyz2cc43",
  "labels": {
    "job": "projects/assessment-2-416823/logs/dataflow.googleapis.com%2Fjob-message"
  },
  "logName": "projects/assessment-2-416823/logs/dataflow.googleapis.com%2Fjob-message",
  "receiveTimestamp": "2024-03-11T22:48:26.677456234Z",
  "resource": {
    "type": "batch"
  },
  "severity": "ERROR",
  "textPayload": "Failed to read the result file : gs://dataflow-staging-asia-east1-866290030587/staging/template_launches/2024-03-11_15_45_52-3889315570494279297/operation_result with error message: (6600fcfb2f6cf71f): Unable to open template file: gs://dataflow-staging-asia-east1-866290030587/staging/template_launches/2024-03-11_15_45_52-3889315570494279297/operation_result..",
  "timestamp": "2024-03-11T22:48:25.584424Z"
}
```

ⓘ No newer entries found matching current filter.

Free trial status: \$485.08 credit and 89 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform. **DISMIS** **ACTIVATE**

Google Cloud **707 Assessment 2** **Search (/) for resources, docs, products, and more** **Q Search**

Dataflow Overview Monitoring Jobs Pipelines Workbench Snapshots SQL Workspace

JOB GRAPH **EXECUTION DETAILS** **JOB METRICS** **COST** **RECOMMENDATIONS**

Logs **HIDE** **▲ 10** **0 1** **+** IMPORT AS PIPELINE **SHARE** **ARCHIVE** **+ TRIGGER** **SEND FEEDBACK**

Job info

Job name: etl-707-assessment-2-mp-171206307-112578914322449922

Job ID: 2024-03-11T18_27-3273611988125618973

Job type: Batch

Job status: Failed

SDK version: Apache Beam SDK for Java 2.53.0

Job region: us-east1

Worker location: us-east1

Current workers: 0

Latest worker status: Worker pool stopped.

Start time: March 12, 2024 at 2:18:28 PM (GMT+13)

Elapsed time: 5 min 30 sec

Encryption type: Google-managed

Dataflow Prime: Disabled

Runner v2: Enabled

Dataflow Shuffle: Enabled

Resource metrics

Current vCPUs: 1

Total vCPUs: 0.027 vCPU hr

Current memory: 3.75 GB

Total memory time: 0.101 GB hr

Current HDD PO: 25 MB

Total HDD PO time: 0.671 GB hr

Current SSD PO: 0 B

Total SSD PO time: 0.008 hr

Total Shuffle data processed: 520 B

Billable Shuffle data processed: 130 B

Note: This pipeline is now using Runner V2. **MORE DETAILS**

ⓘ No newer entries found matching current filter.

Free trial status: \$485.08 credit and 89 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform. **DISMIS** **ACTIVATE**

Google Cloud **707 Assessment 2** **Search (/) for resources, docs, products, and more** **Q Search**

Dataflow Overview Monitoring Jobs Pipelines Workbench Snapshots SQL Workspace

JOB GRAPH **EXECUTION DETAILS** **JOB METRICS** **COST** **RECOMMENDATIONS**

Logs **HIDE** **▲ 4** **0 2** **+** IMPORT AS PIPELINE **SHARE** **ARCHIVE** **+ TRIGGER** **SEND FEEDBACK**

Job info

Job name: etl-707-assessment-2-int-sa-mp-171210128-50997820111998540

Job ID: 2024-03-11T19_22-09-14602097073872112234

Job type: Batch

Job status: Failed

SDK version: Apache Beam SDK for Java 2.53.0

Job region: us-east1

Worker location: us-east1

Current workers: 0

Latest worker status: Worker pool stopped.

Start time: March 12, 2024 at 3:22:08 PM (GMT+13)

Elapsed time: 5 min 41 sec

Encryption type: Google-managed

Dataflow Prime: Disabled

Runner v2: Enabled

Dataflow Shuffle: Enabled

Resource metrics

Current vCPUs: 1

Total vCPUs: 0.034 vCPU hr

Current memory: 3.75 GB

Total memory time: 0.126 GB hr

Current HDD PO: 25 MB

Total HDD PO time: 0.842 GB hr

Current SSD PO: 0 B

Total SSD PO time: 0.008 hr

Total Shuffle data processed: 549 B

Billable Shuffle data processed: 137 B

ⓘ No newer entries found matching current filter.

Data Integration Job Overview

Job Info:

- Job name: etlassessment2
- Job ID: 2024-03-16_16_31-27-1656973020257757595
- Job type: Batch
- Job status: Failed (red)
- SDK version: Apache Beam SDK for Java 2.53.0
 - A newer version of the SDK family exists and updating is recommended. Learn more
- Job region: us-east1
- Worker location: us-east1
- Current workers: -
- Latest worker status: -
 - Start time: March 12, 2024 at 12:31:28 PM GMT+13
 - Elapsed time: 2 min 7 sec
 - Encryption type: Google-managed
 - Dataflow Prime: Disabled
 - Runner v2: Enabled
 - Dataflow Shuffle: Enabled

Resource metrics:

- Current vCPUs: -
- Total vCPU time: - vCPU hr
- Current memory: -
- Total memory time: - GB hr
- Current HDD PD: -
- Total HDD PD time: - GB hr
- Current SSD PD: -
- Total SSD PD time: - GB hr

Labels:

- goog-dataflow-provided-template-name: mongodb_to_bq

Logs:

Severity: Error

Filter: Search all fields and values

EVERY | TIMESTAMP | SUMMARY

```

dataflow.streamingWorkItems.commitWork, dataflow.streamingWorkItems.getData, dataflow.shuffle.read, dataflow.shuffle.write
when accessing projects/assessment-2-416823 as Dataflow worker service account 866298938587-compute@developer.gserviceaccount.com. Learn more at https://cloud.google.com/dataflow/docs/guides/troubleshoot-permissions#validation-failed.

resource: (2)
severity: "ERROR"
textPayload:
  The project's pipeline validation failed for job etlassessment2. To troubleshoot validation, use the Dataflow Service option with the value --enable-preflight-validation=false. Learn more at https://cloud.google.com/dataflow/docs/guides/deploying-a-pipeline#validation] Missing permissions dataflow.workItems.list, dataflow.workItems.update, dataflow.workItems.sendMessage, dataflow.streamingWorkItems.getWork, dataflow.streamingWorkItems.commitWork, dataflow.streamingWorkItems.getData, dataflow.shuffle.read, dataflow.shuffle.write when accessing projects/assessment-2-416823 as Dataflow worker service account 866298938587-compute@developer.gserviceaccount.com. Learn more at https://cloud.google.com/dataflow/docs/guides/troubleshoot-permissions#validation-failed.
  timestamp: "2024-03-11T23:33:36.193434848Z"
}

2024-03-12 12:33:35.293 NZDT Errror processing pipeline;
No newer entries found matching current filter.

```

Job Graph:

Job steps view: Graph view

Job Info:

- Job name: etl707
- Job ID: 2024-03-15_16_57_37-156671690440906147
- Job type: Running (green)
- Job status: Success
- SDK version: Apache Beam SDK for Java 2.54.0
- Job region: asia-southeast1
- Worker location: asia-southeast1
- Current workers: 1
- Latest worker status: Stopping worker pool.
- Start time: March 14, 2024 at 12:57:38 PM GMT+13
- Elapsed time: 6 min 17 sec
- Encryption type: Google-managed
- Dataflow Prime: Disabled
- Runner v2: Enabled
- Dataflow Shuffle: Enabled

Resource metrics:

- Current vCPU: 1
- Total vCPU time: 0.047 vCPU hr
- Current memory: 3.75 GB
- Total memory time: 0.176 GB hr
- Current HDD PD: 25 GB
- Total HDD PD time: 1.173 GB hr
- Current SSD PD: 0 GB
- Total SSD PD time: 0.008 hr
- Total Shuffle data processed: 2 kB
- Billable Shuffle data processed: 513 B

Labels:

- goog-dataflow-provided-template-name: etl707

Logs:

Severity: Error

Filter: Search all fields and values

EVERY | TIMESTAMP | SUMMARY

```

2024-03-14 13:02:28.480 NZDT Error message from worker: org.apache.beam.sdk.util.UserCodeException: java.lang.RuntimeException: Failed to create job with.
2024-03-14 13:02:37.529 NZDT Error message from worker: org.apache.beam.sdk.util.UserCodeException: java.lang.RuntimeException: Failed to create job with.
2024-03-14 13:02:46.332 NZDT Error message from worker: org.apache.beam.sdk.util.UserCodeException: java.lang.RuntimeException: Failed to create job with.
2024-03-14 13:02:54.458 NZDT Error message from worker: org.apache.beam.sdk.util.UserCodeException: java.lang.RuntimeException: Failed to create job with.
2024-03-14 13:02:57.093 NZDT Workflow failed. Causes: 527 Write to Bigquery/BatchLoads/SinglePartitionedShuffle[GroupByKey/ReadWrite to Bigquery/Batch.

Note: This pipeline is now using Runner V2. MORE DETAILS X

```

3. ETL was successful but displayed null values in the PREVIEW table. – To solve this problem, I checked the job settings and changed the user option from NONE to Flatten.

Data Integration

Mira Torritit

The screenshot shows the Google Cloud Dataflow interface. On the left, there's a sidebar with 'Dataflow' selected, followed by 'Overview', 'Monitoring', 'Jobs' (which is currently active), 'Pipelines', 'Workbench', 'Snapshots', and 'SQL Workspace'. The main area has tabs for 'JOB GRAPH', 'EXECUTION DETAILS', 'JOB METRICS', 'COST', and 'RECOMMENDATIONS'. A search bar at the top right says 'Search (/) for resources, docs, products, and more'. Below the tabs is a 'Job steps view' section with 'Graph view' selected. It displays four stages of the pipeline: 'Read Documents' (Succeeded, 6 sec, 2 of 2 stages succeeded), 'UDF' (Succeeded, 0 sec, 1 of 1 stage succeeded), 'Transform to TableRow' (Succeeded, 4 sec, 1 of 1 stage succeeded), and 'Write to BigQuery' (Succeeded, 15 sec, 20 of 20 stages succeeded). To the right of the graph is a 'Job info' panel with detailed information about the job, including its name, ID, type, status, region, and various metrics like vCPUs, memory, and disk usage. A 'Resource metrics' section also provides detailed resource utilization data.

The screenshot shows the Google Cloud BigQuery interface. At the top, it says 'Free trial status: \$485.07 credit and 89 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.' The main area has tabs for 'SCHEMA', 'DETAILS', 'PREVIEW', 'LINEAGE', 'DATA PROFILE', and 'DATA QUALITY'. The 'amazon_sales_report' table is selected. The 'PREVIEW' tab is active, showing a table with columns: Row, id, stop-postal-code, ship-country, promotion-ids, B2B, id, and source_data. There are six rows of data. The 'source_data' column contains JSON objects. The 'DETAILS' tab shows the schema: id (string), stop-postal-code (string), ship-country (string), promotion-ids (string), B2B (string), id (string), and source_data (string). The 'SCHEMA' tab shows the same schema. The 'LINEAGE' tab shows the table's lineage. The 'DATA PROFILE' and 'DATA QUALITY' tabs are also visible. On the left, there's an 'Explorer' sidebar with a tree view of resources, including 'assessment-2-416823' and 'amazon_sales_report'. The bottom of the screen shows a 'Job history' section with a refresh button.

Part B: Big Data Analysis and Application of Engineering Techniques

Task 1: Data Ingestion Pipeline

1. Download File from Kaggle.

The screenshot shows a Kaggle dataset page for 'Market Basket Analysis'. The page title is 'Market Basket Analysis' with a subtitle 'Analyzing Consumer Behaviour Using MBA Association Rule Mining'. On the right, there's a thumbnail image of a shopping cart filled with groceries. The left sidebar has a 'Datasets' section selected. The main content area includes sections for 'About Dataset', 'Market Basket Analysis', 'Introduction', and 'An Example of Association Rules'. A sidebar on the right lists 'Usability', 'License', 'Expected update frequency', and 'Tags'.

2. I created a new project in GCP, 707 Assessment 2—Part B. I also enabled Cloud Dataproc API and created a cluster on Compute Engine.

The screenshot shows the GCP interface with the 'Dataproc' service selected. The left sidebar shows 'Clusters' under 'Jobs on Clusters'. The main pane displays a table for a cluster named 'cluster-14f9'. The table columns include Name, Status, Region, Zone, Total worker nodes, Flexible VMs?, Scheduled deletion, Cloud Storage staging bucket, Created, and Labels. The cluster is running in the 'asia-southeast1-b' zone with 2 worker nodes. It was created on March 15, 2024, at 8:11:57 PM.

- Created a new Bucket named 707dataset_kaggle and uploaded the two datasets from Kaggle (Source GCP Storage Bucket)

Bucket details for 707dataset_kaggle

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Object retention	Retain until time
Assignment-1_Data.csv	39.3 MB	text/csv	Mar 15, 2024, 6:20:52 PM	Standard	Mar 15, 2024, 6:20:52 PM	Not public	—	Google-managed	—	—
data.csv	43.5 MB	text/csv	Mar 15, 2024, 6:26:42 PM	Standard	Mar 15, 2024, 6:26:42 PM	Not public	—	Google-managed	—	—

- Google Cloud console's built-in SSH interface

Cluster details for cluster-14f9

SSH-in-browser

Establishing connection to SSH server...

Authorize

Allow SSH-in-browser to connect to VMs.

Authorize Cancel

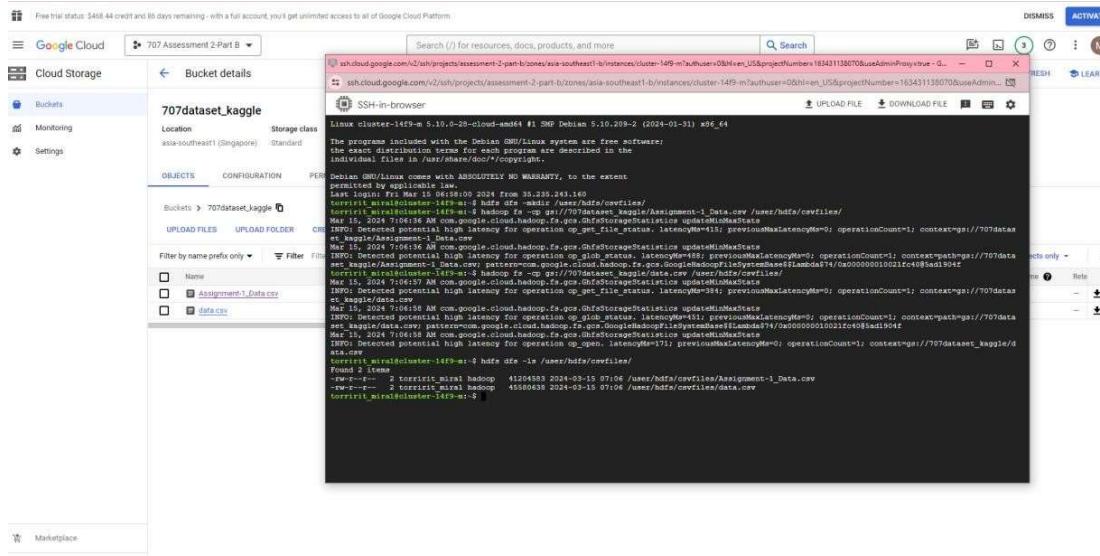
Cluster details for cluster-14f9

SSH-in-browser

```
Linux cluster-14f9-m 5.10.0-28-cloud-amd64 #1 SMP Debian 5.10.209-2 (2024-01-31) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@cluster-14f9-m:~# ls
archive
root@cluster-14f9-m:~# cd archive
root@cluster-14f9-m:~/archive# ls
47.zip
root@cluster-14f9-m:~/archive#
```

- Creating a destination folder (HDFS)



Task 2: Implementing Data Storage

MongoDB Atlas was chosen because it is user-friendly. The database can adjust easily to different formats as it is unstructured. The data was loaded in Python using the Pandas library and loaded into MongoDB.

```
import pandas as pd
import numpy as np

data_df = pd.read_csv('data.csv')

from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi

def get_mongodbclient():

    uri = 'mongodb+srv://707user:ADzgnRDrEQj3Fqfd@707assessment.eld0gap.mongodb.net/?retryWrites=true&w=majority&appName=707Asses

    # Create a new client and connect to the server
    client = MongoClient(uri, server_api=ServerApi('1'))

    # Send a ping to confirm a successful connection
    try:
        client.admin.command('ping')
        print("Pinged your deployment. You successfully connected to MongoDB!")
        return client
    except Exception as e:
        print(e)
        return None

# Connect to MongoDB Atlas
client = get_mongodbclient()
# Initiate an instance of the database
db = client['Assessment2_707_Part_B']

Pinged your deployment. You successfully connected to MongoDB!
```

Inserted 541,909 Documents

```
In [32]: # f) Import the dataset into a table or collection within the database.
# In this case, import the data frame to the collection
data_records = data_df.to_dict(orient='records')
collection = db.data
collection.insert_many(data_records)

Out[32]: InsertManyResult([ObjectId('65f40059d0a9120a5febfd4d'), ObjectId('65f40059d0a9120a5febfb4d'), ObjectId('65f40059d0a9120a5febfd4f'), ObjectId('65f40059d0a9120a5febfd50'), ObjectId('65f40059d0a9120a5febfd51'), ObjectId('65f40059d0a9120a5febfd52'), ObjectId('65f40059d0a9120a5febfd53'), ObjectId('65f40059d0a9120a5febfd54'), ObjectId('65f40059d0a9120a5febfd55'), ObjectId('65f40059d0a9120a5febfd56'), ObjectId('65f40059d0a9120a5febfd57'), ObjectId('65f40059d0a9120a5febfd58'), ObjectId('65f40059d0a9120a5febfd59'), ObjectId('65f40059d0a9120a5febfd5a'), ObjectId('65f40059d0a9120a5febfd5b'), ObjectId('65f40059d0a9120a5febfd5c'), ObjectId('65f40059d0a9120a5febfd5d'), ObjectId('65f40059d0a9120a5febfd5e'), ObjectId('65f40059d0a9120a5febfd5f'), ObjectId('65f40059d0a9120a5febfd61'), ObjectId('65f40059d0a9120a5febfd62'), ObjectId('65f40059d0a9120a5febfd63'), ObjectId('65f40059d0a9120a5febfd64'), ObjectId('65f40059d0a9120a5febfd65'), ObjectId('65f40059d0a9120a5febfd66'), ObjectId('65f40059d0a9120a5febfd67'), ObjectId('65f40059d0a9120a5febfd68'), ObjectId('65f40059d0a9120a5febfd69'), ObjectId('65f40059d0a9120a5febfd6a'), ObjectId('65f40059d0a9120a5febfd6b'), ObjectId('65f40059d0a9120a5febfd6c'), ObjectId('65f40059d0a9120a5febfd6d'), ObjectId('65f40059d0a9120a5febfd6e'), ObjectId('65f40059d0a9120a5febfd6f'), ObjectId('65f40059d0a9120a5febfd6g'), ObjectId('65f40059d0a9120a5febfd6h'), ObjectId('65f40059d0a9120a5febfd6i'), ObjectId('65f40059d0a9120a5febfd6j'), ObjectId('65f40059d0a9120a5febfd6k'), ObjectId('65f40059d0a9120a5febfd6l'), ObjectId('65f40059d0a9120a5febfd6m'), ObjectId('65f40059d0a9120a5febfd6n'), ObjectId('65f40059d0a9120a5febfd6o'), ObjectId('65f40059d0a9120a5febfd6p'), ObjectId('65f40059d0a9120a5febfd6q'), ObjectId('65f40059d0a9120a5febfd6r'), ObjectId('65f40059d0a9120a5febfd6s'), ObjectId('65f40059d0a9120a5febfd6t'), ObjectId('65f40059d0a9120a5febfd6u'), ObjectId('65f40059d0a9120a5febfd6v'), ObjectId('65f40059d0a9120a5febfd6w'), ObjectId('65f40059d0a9120a5febfd6x'), ObjectId('65f40059d0a9120a5febfd6y'), ObjectId('65f40059d0a9120a5febfd6z'), ObjectId('65f40059d0a9120a5febfd70'), ObjectId('65f40059d0a9120a5febfd71'), ObjectId('65f40059d0a9120a5febfd72'), ObjectId('65f40059d0a9120a5febfd73'), ObjectId('65f40059d0a9120a5febfd74'), ObjectId('65f40059d0a9120a5febfd75'), ObjectId('65f40059d0a9120a5febfd76'), ObjectId('65f40059d0a9120a5febfd77'), ObjectId('65f40059d0a9120a5febfd78'), ObjectId('65f40059d0a9120a5febfd79'), ObjectId('65f40059d0a9120a5febfd7a'), ObjectId('65f40059d0a9120a5febfd7b'), ObjectId('65f40059d0a9120a5febfd7c'), ObjectId('65f40059d0a9120a5febfd7d'), ObjectId('65f40059d0a9120a5febfd7e'), ObjectId('65f40059d0a9120a5febfd7f'), ObjectId('65f40059d0a9120a5febfd80'), ObjectId('65f40059d0a9120a5febfd81'), ObjectId('65f40059d0a9120a5febfd82'), ObjectId('65f40059d0a9120a5febfd83'), ObjectId('65f40059d0a9120a5febfd84'), ObjectId('65f40059d0a9120a5febfd85'), ObjectId('65f40059d0a9120a5febfd86'), ObjectId('65f40059d0a9120a5febfd87'), ObjectId('65f40059d0a9120a5febfd88'), ObjectId('65f40059d0a9120a5febfd89'), ObjectId('65f40059d0a9120a5febfd89')])
```

The screenshot shows the MIRA's ORO interface with the following details:

- VERSION:** 7.0.6
- REGION:** GCP Singapore (asia-southeast1)
- Collections:** Assessment2_707_Part_B
- Storage Size:** 29.98MB
- Logical Data Size:** 15.27MB
- Total Documents:** 54909
- Indexes Total Size:** 15.04MB
- Find:** Type a query: { Field: 'value' }
- Filter:** Filter ID
- Options:** Refresh, Insert Document, Reset, Apply
- QUERY RESULTS: 1-20 OF MANY**
- Document Preview:** Shows two documents with their IDs, invoice numbers, stock codes, descriptions, quantities, and other metadata.

Performed Query to get the 541,909 Documents.

```
In [33]: # g) Retrieve and display records or documents from the table or collection.
# using collection.find to get all the records from adidas_collection
collection = db.data
data = collection.find()

new_data_df = pd.DataFrame(list(data))
new_data_df
```

	_id	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	65f40059d0a9120a5febfd4c	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	65f40059d0a9120a5febfd4d	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	65f40059d0a9120a5febfd4e	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	65f40059d0a9120a5febfd4f	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	65f40059d0a9120a5febfd50	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
...
541904	65f4005bd0a9120a5ff4421c	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	12/9/2011 12:50	0.85	12680.0	France
541905	65f4005bd0a9120a5ff4421d	581587	22899	CHILDRENS APRON DOLLY GIRL	6	12/9/2011 12:50	2.10	12680.0	France
541906	65f4005bd0a9120a5ff4421e	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12/9/2011 12:50	4.15	12680.0	France
541907	65f4005bd0a9120a5ff4421f	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12/9/2011 12:50	4.15	12680.0	France
541908	65f4005bd0a9120a5ff44220	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12/9/2011 12:50	4.95	12680.0	France

541909 rows × 9 columns

Performed sorting for the 541,909 Documents

```
In [34]: # h) Sort the records or documents based on a given condition.
# Sort by Invoice Date
sorted_new_data_df = new_data_df.sort_values(by='InvoiceDate', ascending=False)
sorted_new_data_df
```

	_id	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
332571	65f4005ad0a9120a5ff11067	566079	22923	FRIDGE MAGNETS LES ENFANTS ASSORTED	24	9/9/2011 9:52	0.85	17593.0	United Kingdom
332544	65f4005ad0a9120a5ff1104c	566079	23403	LETTER HOLDER HOME SWEET HOME	4	9/9/2011 9:52	3.75	17593.0	United Kingdom
332551	65f4005ad0a9120a5ff11053	566079	22396	MAGNETS PACK OF 4 RETRO PHOTO	24	9/9/2011 9:52	0.39	17593.0	United Kingdom
332550	65f4005ad0a9120a5ff11052	566079	22400	MAGNETS PACK OF 4 HOME SWEET HOME	24	9/9/2011 9:52	0.39	17593.0	United Kingdom
332549	65f4005ad0a9120a5ff11051	566079	20838	FRENCH LATTICE CUSHION COVER	12	9/9/2011 9:52	0.85	17593.0	United Kingdom
...
50831	65f40059d0a9120a5fec3db	540561	22743	MAKE YOUR OWN FLOWERPOWER CARD KIT	6	1/10/2011 10:32	2.95	13004.0	United Kingdom
50809	65f40059d0a9120a5fec3c5	540561	22343	PARTY PIZZA DISH RED RETROSPOT	24	1/10/2011 10:32	0.21	13004.0	United Kingdom
50808	65f40059d0a9120a5fec3c4	540560	21589		NaN	-14	1/10/2011 10:08	0.00	NaN
50807	65f40059d0a9120a5fec3c3	C540559	21888		BINGO SET	-4	1/10/2011 10:07	3.75	NaN
50806	65f40059d0a9120a5fec3c2	540558	21258		?	-29	1/10/2011 10:04	0.00	NaN

541909 rows × 9 columns

Counted the number of documents using the Python Application

```
In [35]: # i) Count the number of records or documents present in the table or collection.
collection = db.data
document_count = collection.count_documents({})
print(f'Total Documents: {document_count}'')
```

Total Documents in adidas_collection: 541909

Performed Grouping

```
In [36]: # j) Perform grouping operations on records or documents within the table or collection.
collection = db.data
grouping = [
    {"$group": {
        "_id": "$Country",
        "count" : {"$sum" : 1}
    }
}
result = list(collection.aggregate(grouping))
print(result)
```

[{"_id": "Australia", "count": 1259}, {"_id": "EIRE", "count": 8196}, {"_id": "Greece", "count": 146}, {"_id": "USA", "count": 291}, {"_id": "Iceland", "count": 182}, {"_id": "Canada", "count": 151}, {"_id": "Israel", "count": 297}, {"_id": "Unspecified", "count": 446}, {"_id": "RSA", "count": 58}, {"_id": "Malta", "count": 127}, {"_id": "France", "count": 8557}, {"_id": "Sweden", "count": 462}, {"_id": "Portugal", "count": 1519}, {"_id": "Spain", "count": 2533}, {"_id": "Lithuania", "count": 35}, {"_id": "Netherlands", "count": 2371}, {"_id": "Denmark", "count": 389}, {"_id": "Belgium", "count": 2069}, {"_id": "Bahrain", "count": 19}, {"_id": "Saudi Arabia", "count": 341}, {"_id": "Hong Kong", "count": 288}, {"_id": "United Kingdom", "count": 495478}, {"_id": "Channel Islands", "count": 758}, {"_id": "Italy", "count": 803}, {"_id": "Czech Republic", "count": 30}, {"_id": "Switzerland", "count": 2002}, {"_id": "European Community", "count": 61}, {"_id": "Singapore", "count": 229}, {"_id": "Cyprus", "count": 622}, {"_id": "Finland", "count": 695}, {"_id": "Japan", "count": 358}, {"_id": "Lebanon", "count": 45}, {"_id": "Germany", "count": 9495}, {"_id": "United Arab Emirates", "count": 68}, {"_id": "Brazil", "count": 32}, {"_id": "Austria", "count": 401}]

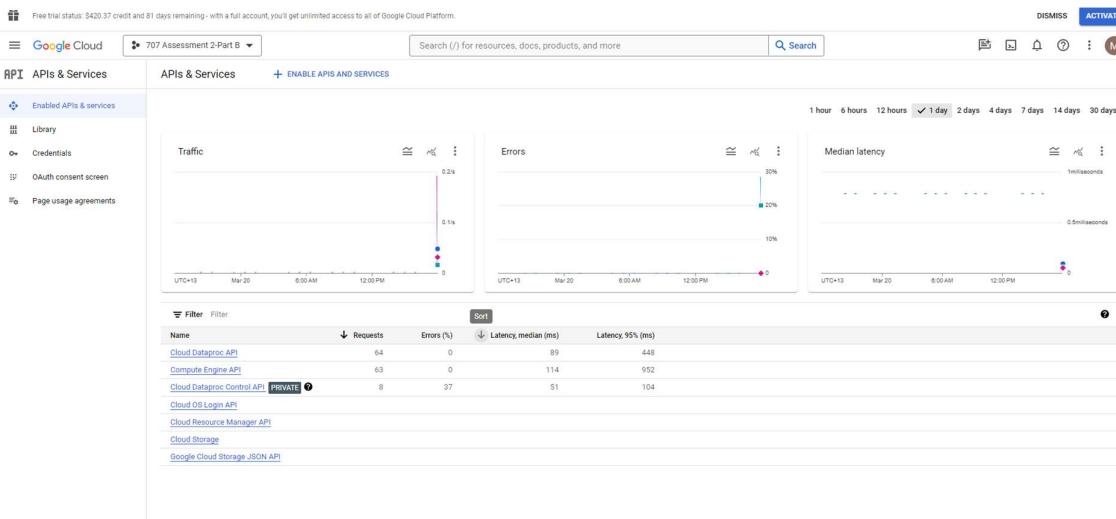
Task 3: Building a real-time data clustering system using Apache Spark

In this activity, I still used the GCP to utilize my free credits. Also, the following are the reasons for considering Data Proc on GCP in managing real-time data, from Facebook Graph API.

1. Scalability: Can smoothly adjust its scale up or down based on the amount of data being processed. Dealing with fluctuating volumes of data from services, like the Facebook Graph API requires processing resources to efficiently handle the workload.
2. Managed Service: This is fully managed by Google Cloud taking care of tasks like infrastructure management such as provisioning, configuration, and monitoring of computer instances. This allows you to concentrate on developing your streaming application without being burdened by responsibilities.
3. Integration with GCP Services: Seamlessly integrates with Google Cloud services like Big Query for data warehousing Pub/Sub for messaging and Dataflow for stream processing. This integration facilitates the creation of end-to-end data pipelines—from ingestion to analysis—in a manner.
4. Support for Spark Streaming: When you use Apache Spark for processing streaming data DataProc offers support for Spark Streaming that's ideal, for managing real-time data streams. With familiar APIs used in batch processing, Spark Streaming simplifies the development of streaming applications.
5. Resource Optimization: Gives you the ability to enhance your resource utilization and manage expenses by defining the type and quantity of machines (VMs) required for your processing activities. This adaptability empowers you to strike a balance between performance and cost efficiency according to your needs. In essence, leveraging DataProc on GCP for streaming data from the Facebook Graph API offers an expandable solution with operational burden enabling you to concentrate on extracting valuable insights, from your streaming data.

Steps on building the real-time data clustering (with an FB account)

1. Go to Data proc clusters, select the cluster, or create a new cluster (essential details to add: Jupyter). To enable the Jupyter interface, go to the navigation menu, API & services, and click + enable APIS and Services.



In the search bar, type cloud resource manager API

The screenshot shows the Google Cloud API Library search results for 'cloud resource manager api'. The search bar contains 'cloud resource manager api'. The results are displayed in a grid format:

- Maps** (VIEW ALL (23))
 - Maps SDK for Android (Google) - Maps for your native Android app.
 - Maps SDK for iOS (Google) - Maps for your native iOS app.
 - Maps JavaScript API (Google) - Maps for your website.
 - Places API (Google Enterprise API) - Get detailed information about 100 million places.
 - Roads API (Google Enterprise API) - High-stability APIs. Ready for enterprise use. Support options available. LEARN MORE
- Machine learning** (VIEW ALL (13))
 - DialogFlow API (Google Enterprise API) - Builds conversational interfaces.
 - Cloud Vision API (Google Enterprise API) - Image Content Analysis.
 - Cloud Natural Language API (Google Enterprise API) - Provides natural language understanding technologies, such as sentiment analysis, entity...
 - Cloud Speech-to-Text API (Google Enterprise API) - Speech recognition.
 - Cloud Translation API (Google Enterprise API) - Integrates text translation into your website or application.
 - AI Platform Training & Prediction API (Google Enterprise API) - An API to enable creating and using machine learning models.

Click Cloud Resource Manager API to Enable

The screenshot shows the Google Cloud API Library search results for 'cloud resource manager api'. The search bar contains 'cloud resource manager api'. One result is selected:

- Cloud Resource Manager API** (Google Enterprise API)

Creates, reads, and updates metadata for Google Cloud Platform resource containers.

Enabled Cloud Resource Manager API

The screenshot shows the Google Cloud Platform API library interface. At the top, there's a banner indicating a free trial status with \$420.37 credit and 81 days remaining. Below the banner, the navigation bar includes 'Google Cloud' and '707 Assessment 2-Part B'. A search bar and various navigation icons are also present.

The main content area displays the 'Cloud Resource Manager API' page. It features a circular icon with a play button, the API name, and a brief description: 'Creates, reads, and updates metadata for Google Cloud Platform resource containers.' Below this, there are buttons for 'MANAGE', 'TRY THIS API', and 'API Enabled'.

Below the main description, there are tabs for 'OVERVIEW', 'DOCUMENTATION', and 'RELATED PRODUCTS'. The 'OVERVIEW' tab is selected. Under 'Additional details', it lists the type as 'SaaS & APIs', last update date as '7/22/22', category as 'Google Enterprise APIs', and service name as 'cloudresourcemanager.googleapis.com'.

Further down, sections for 'Tutorials and documentation' and 'Terms of Service' are visible. The 'Terms of Service' section contains a link to the Google Cloud Platform terms of service.

Then go back to the cluster: cluster-with-spark.

The screenshot shows the Google Cloud Platform Dataproc Clusters page. The left sidebar has sections for 'Jobs on Clusters' (Clusters, Jobs, Workflows, Autoscaling policies), 'Serverless' (Batches, Interactive), 'Metastore Services' (Metastore, Federation), and 'Utilities' (Component exchange, Workbench). The main content area shows a table of clusters:

Name	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion	Cloud Storage staging bucket	Created	Labels
cluster-14f9	Stopped	asia-southeast1	asia-southeast1-b	2	No	Off	dataproc-staging-asia-southeast1-163431138070j4sfclc3z	Mar 15, 2024, 6:11:57PM	goog-datap... enabled
cluster-with-spark	Running	asia-southeast1	asia-southeast1-c	2	No	Off	dataproc-staging-asia-southeast1-163431138070j4sfclc3z	Mar 15, 2024, 10:47:30PM	goog-datap... enabled

Under web interfaces, select Jupyter.

For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See <https://cloud.google.com/compute/docs/disks/performance> for information on disk I/O performance.

Creating a cluster

1. Go to Dataproc , + create a cluster, choose cluster on compute engine

Free trial status: \$419.75 credit and 81 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

Create a Dataproc cluster on Compute Engine

Jobs on Clusters

- Clusters** (selected)
- Jobs
- Workflows
- Autoscaling policies

Serverless

- Batches
- Interactive

Metastore Services

- Metastore
- Federation

Utilities

- Component exchange
- Workbench

Release Notes

CREATE CANCEL

EQUIVALENT COMMAND LINE

Configure nodes (optional)

Change node compute and storage capabilities.

Customize cluster (optional)

Add cluster properties, features, and actions.

Manage security (optional)

Change access, encryption, and security settings.

General purpose (selected) Compute optimized Memory optimized GPUs

Machine types for common workloads, optimized for cost and flexibility

Series: E2 CPU platform selection based on availability

Machine type: e2-standard-2 (2 vCPU, 1 core, 8 GB memory)

vCPU: 2 **Memory**: 8 GB

CPU PLATFORM AND GPU

Primary disk size: 100 GB **Primary disk type**: Balanced Persistent Disk

Number of local SSDs: x 3750B **Local SSD Interface**: Local SSD Interface

Worker nodes

Each contains a YARN NodeManager and a HDFS DataNode. HDFS replication factor is 2.

General purpose (selected) Compute optimized Memory optimized GPUs

Machine types for common workloads, optimized for cost and flexibility

Series: E2 CPU platform selection based on availability

Machine type: e2-standard-2 (2 vCPU, 1 core, 8 GB memory)

vCPU: 2 **Memory**: 8 GB

CPU PLATFORM AND GPU

Number of worker nodes: 1

Free trial status: \$419.75 credit and 81 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

Create a Dataproc cluster on Compute Engine

Jobs on Clusters

- Clusters** (selected)
- Jobs
- Workflows
- Autoscaling policies

Serverless

- Batches
- Interactive

Metastore Services

- Metastore
- Federation

Utilities

- Component exchange
- Workbench

Release Notes

CREATE CANCEL

EQUIVALENT COMMAND LINE

Set up cluster

Begin by providing basic information.

Configure nodes (optional)

Change node compute and storage capabilities.

Customize cluster (optional)

Add cluster properties, features, and actions.

Manage security (optional)

Change access, encryption, and security settings.

Internal IP only

Configure all instances to have only internal IP addresses. [Learn more](#)

Labels

A list of key/value pairs to attach to the cluster for tracking.

+ ADD LABELS

Cluster properties

Use cluster properties to add or modify configuration files when creating a cluster.

+ ADD PROPERTIES

Initialization actions

Use initialization actions to customize settings, install applications, or make other modifications to your cluster. Select scripts or executables that Cloud Dataproc will run when provisioning your cluster.

+ ADD INITIALIZATION ACTION

Custom cluster metadata

Add custom metadata to cluster instances. [Learn more](#)

+ ADD METADATA

Scheduled deletion

Use Scheduled Deletion to help avoid incurring Google Cloud charges for an inactive cluster. [Learn more](#)

Delete on a fixed time schedule

Delete after a cluster idle time period without submitted jobs

Cloud Storage staging bucket

gs://dataproc-staging-us-central1-163431138070-x04bf4de

App form - torrit...

Free trial status: \$419.75 credit and 81 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

Clusters

CREATE CLUSTER **REFRESH** **START** **STOP** **DELETE** **REGIONS** **+ 5 RECOMMENDED ALERTS** **SHOW INFO PANEL** **LEARN**

Jobs on Clusters

- Clusters** (selected)
- Jobs
- Workflows
- Autoscaling policies

Serverless

- Batches
- Interactive

Metastore Services

- Metastore
- Federation

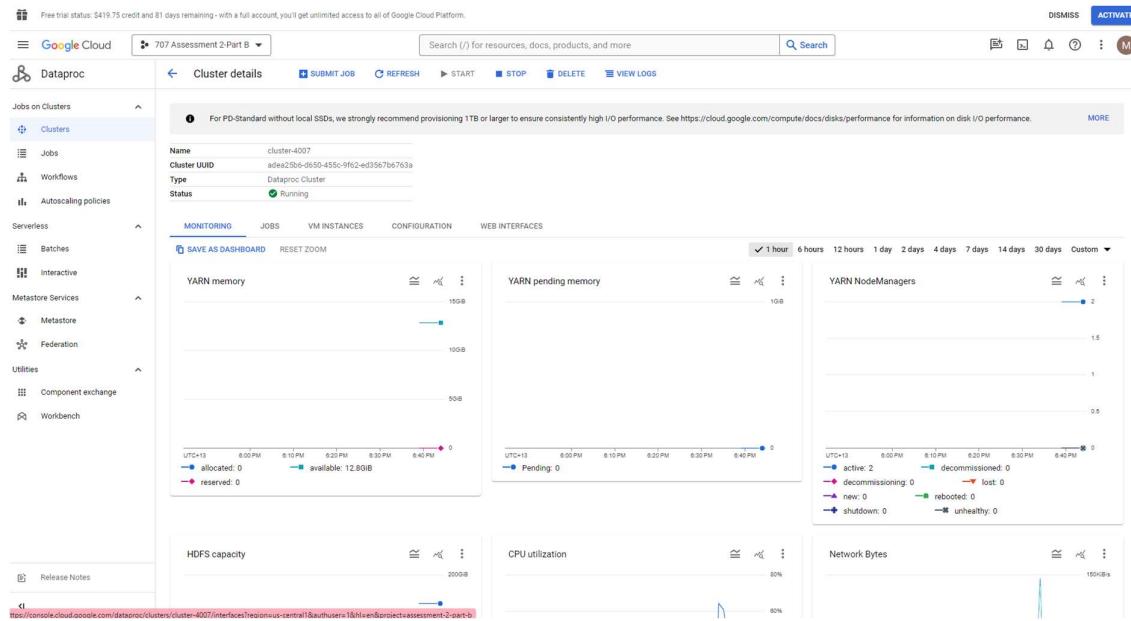
Utilities

- Component exchange
- Workbench

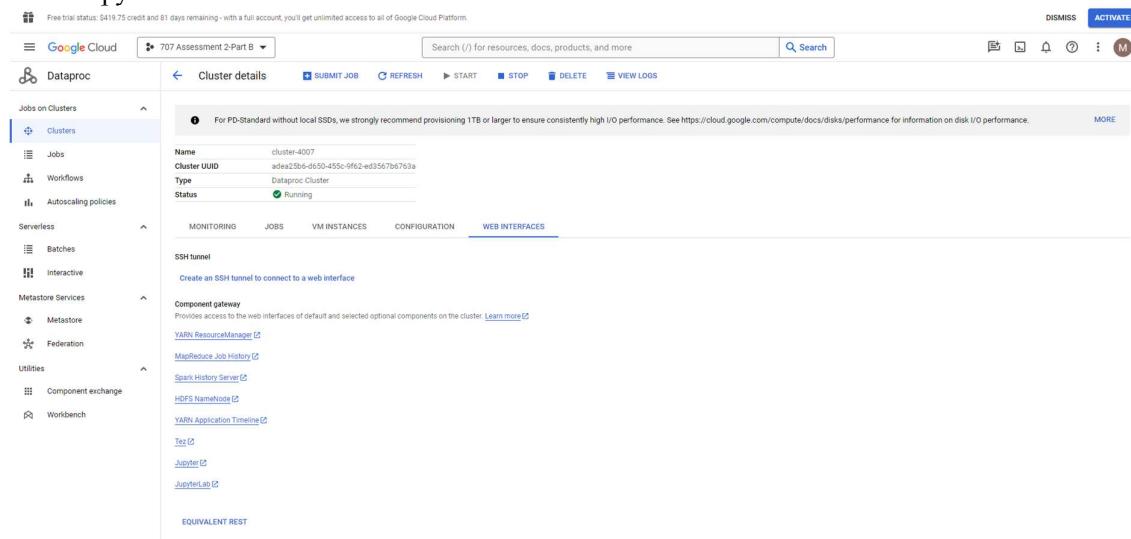
Filter Search clusters, press Enter

Name	Status	Region	Zone	Total worker nodes	Flexible VMs?	Scheduled deletion	Cloud Storage staging bucket	Created	Labels
cluster-14f9	Stopped	asia-southeast1	asia-southeast1-b	2	No	Off	dataproc-staging-asia-southeast1-163431138070-49fc3z	Mar 15, 2024, 6:11:57 PM	goog-dataproc..._enabled
cluster-4007	Running	us-central1	us-central1-b	2	No	Off	dataproc-staging-us-central1-163431138070-x04bf4de	Mar 20, 2024, 6:32:40 PM	goog-dataproc..._enabled

Click the web interface.

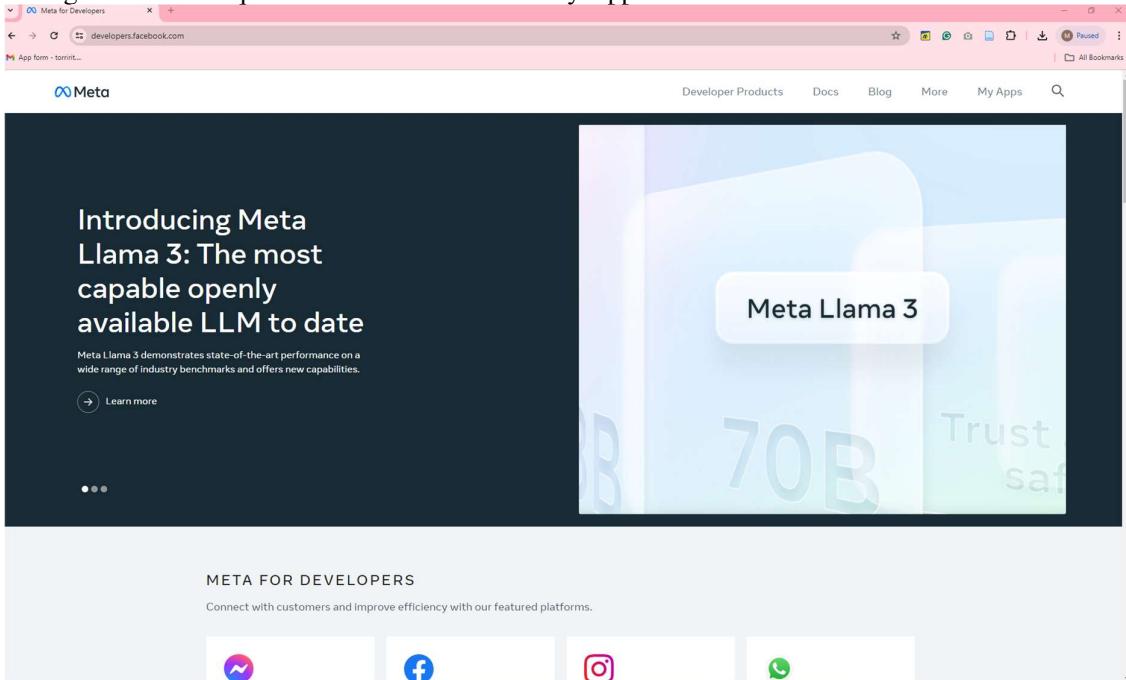


Click Jupyter



- Generating Access Token from Facebook – this allows you to connect to the Facebook account.

1. Register to Developers.facebook.com. Select My Apps



2. Click Create Apps

The screenshot shows the Meta for Developers Apps dashboard. On the left, there's a sidebar with a 'Filter by' section containing 'All Apps' (selected), 'Archived', and 'Required actions'. Below that is a 'Business portfolio' dropdown set to 'No business portfolio selected'. The main area has a large illustration of a person sitting at a desk in front of a computer. A central message says 'No apps yet' with the subtext 'To get started, create your first app.' and a prominent green 'Create App' button.

3. Select Authenticate and request data from users with Facebook Login, click next

The screenshot shows the 'Create an app' wizard. In the 'Add use case' section, 'Authenticate and request data from users with Facebook Login' is selected (indicated by a blue border). Other options like 'Launch a game on Facebook' and 'Other' are also shown. At the bottom right of the wizard, there's a 'Next' button.

4. Select No, I'm not building a game, click next

The screenshot shows the 'Create an app' wizard. In the 'Login type' section, 'No, I'm not building a game' is selected (indicated by a blue border). Other options like 'Yes, I'm building a game' are also shown. At the bottom right of the wizard, there's a 'Next' button.

5. Add an app name (I named it Assessment 2-Part B), then click Create App

The screenshot shows the 'Create an app' form on the Facebook Developer Portal. The 'Add app name' field is filled with 'Assessment 2 - Part B'. The 'App contact email' field contains 'lun2isolation@yahoo.com'. The 'Business portfolio' section is optional and shows 'No Business Manager account selected'. At the bottom right are 'Previous' and 'Create app' buttons.

6. Once created, click Use cases – to allow you to access certain data in the Facebook account

The screenshot shows the Facebook Developer Portal Dashboard. The 'Use cases' link in the sidebar is highlighted. The main dashboard shows steps for creating and publishing the app, with '1. Customize this app' expanded to show options like 'Customize adding a Facebook Login button' and 'Explore and add more use cases'.

7. Select Use additional Facebook user data for personalization

The screenshot shows the 'Use cases' section of the Meta for Developers dashboard. On the left, there's a sidebar with links like 'App settings', 'App roles', and 'Alert Inbox'. The main area lists several use cases: 'Authenticates and request data from users with Facebook Login' (selected), 'Authentication and account creation', 'Available use cases' (with a note about common use cases), 'Use additional Facebook user data for personalization' (highlighted with a red border), 'Track engagement with Meta App Events', and 'Get real-time notifications with Webhooks'. Each use case has a 'Customize' and a 'Delete' button.

8. Below are the items you can access, just click add to allow them.

The screenshot shows the 'Permissions' section of the 'Customize' page for a specific use case. It lists 15 different permissions, each with a brief description and an 'Add' button. The permissions are: user_age_range, user_birthday, user_friends, user_gender, user_hometown, user_likes, user_link, user_location, user_photos, user_posts, and user_videos. Each permission row includes a 'Full Description' and 'Requirements' link.

Permission	Description	Action
user_age_range	The user_age_range permission allows your app to access a person's age range as listed in their Facebook profile.	Add
user_birthday	The user_birthday permission allows your app to read a person's birthday as listed in their Facebook profile.	Add
user_friends	The user_friends permission allows your app to get a list of a person's friends using that app.	Add
user_gender	The user_gender permission allows your app to read a person's gender as listed in their Facebook profile.	Add
user_hometown	The user_hometown permission allows your app to read a person's hometown location from their Facebook profile.	Add
user_likes	The user_likes permission allows your app to read a list of all Facebook Pages that a user has liked.	Add
user_link	The user_link permission allows your app to access the Facebook profile URL of the person using your app.	Add
user_location	The user_location permission allows your app to read the city name as listed in the location field of a person's Facebook profile.	Add
user_photos	The user_photos permission allows your app to read the photos a person has uploaded to Facebook.	Add
user_posts	The user_posts permission allows your app to access the posts that a user has made on their timeline.	Add
user_videos	The user_videos permission allows your app to read a list of videos uploaded by a person.	Add

Note: I only accessed the likes and posts, but you can add more depending on your requirements.

The screenshot shows a list of permissions under the 'Meta for Developers' section. The permissions listed are:

- user_gender**: The user_gender permission allows your app to read a person's gender as listed in their Facebook profile. Status: Ready for testing.
- user_hometown**: The user_hometown permission allows your app to read a person's hometown location from their Facebook profile. Status: Ready for testing.
- user_likes**: The user_likes permission allows your app to read a list of all Facebook Pages that a user has liked. Status: Ready for testing.
- user_link**: The user_link permission allows your app to access the Facebook profile URL of the person using your app.
- user_location**: The user_location permission allows your app to read the city name as listed in the location field of a person's Facebook profile.
- user_photos**: The user_photos permission allows your app to read the photos a person has uploaded to Facebook.
- user_posts**: The user_posts permission allows your app to access the posts that a user has made on their timeline. Status: Ready for testing.
- user_videos**: The user_videos permission allows your app to read a list of videos uploaded by a person.

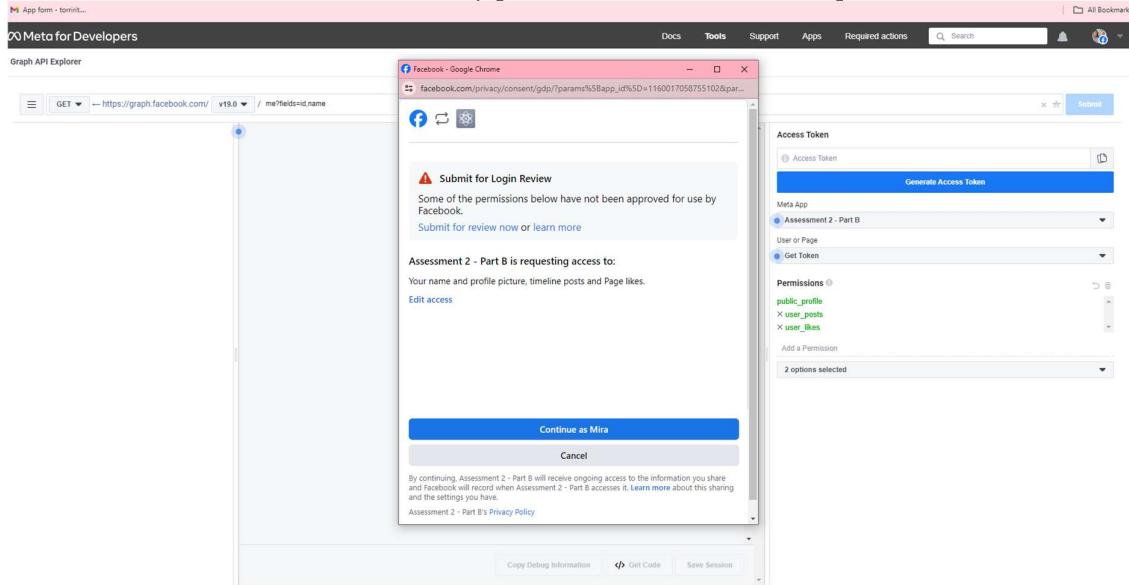
In the top menu bar, hover Tools and select Graph API Explorer. Choose the Meta App created, the action you are going to do (Get Token) and Add permissions, to access the two permissions you added from Use cases. Then click Generate Access Token.

The screenshot shows the Graph API Explorer interface. The URL is set to `https://graph.facebook.com/v19.0/me?fields=id,name`. The 'Access Token' section is expanded, showing the following configuration:

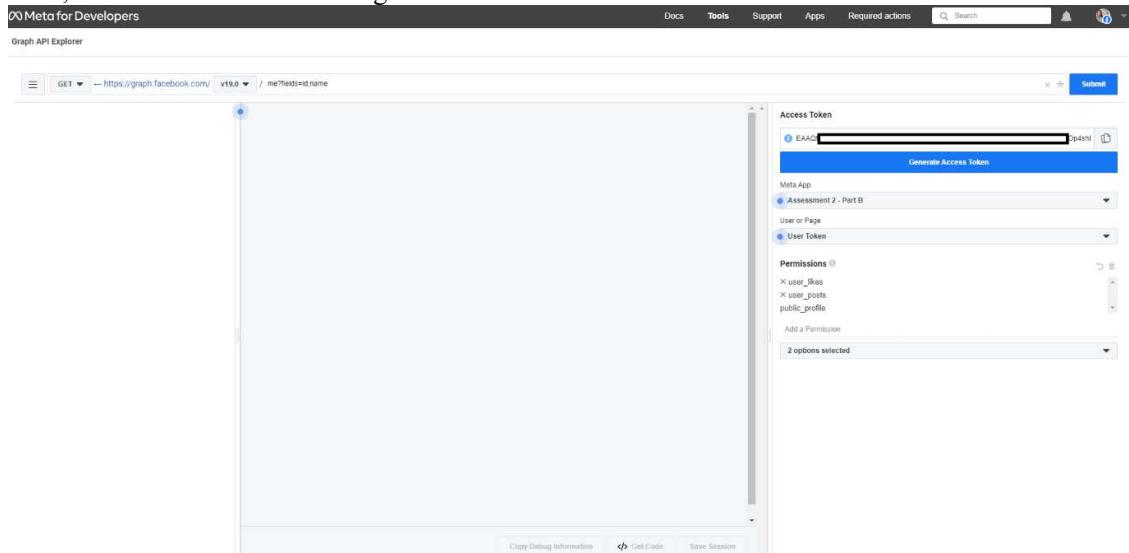
- Meta App: Assessment 2 - Part B
- User or Page: Get Token
- Permissions:
 - public_profile (selected)
 - user_likes
 - user_posts

The 'Add a Permission' section shows the selected permissions: user_likes and user_posts. At the bottom, there are buttons for 'Copy Debug Information', 'Get Code', and 'Save Session'.

Click on Continue as Mira (since I used my personal FB account) to allow permission



Now, the access Token has been generated.



Click submit to test the token if it is working and to get the user ID.

The screenshot shows the Facebook Graph API Explorer interface. The URL in the address bar is `https://graph.facebook.com/v19.0/me?fields=id,name`. The response pane displays a JSON object with fields `id` and `name`, both of which have been selected. The `name` field contains the value "Mira Myres". The right sidebar shows an access token and its permissions: `x_user_likes`, `x_user_posts`, and `public_profile`.

Then Go to GCP, Dataproc, Cluster, and Jupyter and copy the access token and ID to the Python script, then run the script using the request library to access the graph.facebook.com

The screenshot shows a Jupyter Notebook cell with the code:

```

In [5]: import requests
access_token = "REDACTED"
user_name = "REDACTED"
url = f"https://graph.facebook.com/{user_name}?fields=id,name,posts,likes&access_token={access_token}"
response = requests.get(url)
data = response.json()
print(data)

posts = data['posts']
num_of_posts = len(posts['data'])
likes = data['likes']
num_of_likes = len(likes['data'])
print(f"Number of Posts: {num_of_posts}")
print(f"Number of Likes: {num_of_likes}")
print(posts['data'])

```

The output pane shows the JSON response from the Facebook API, listing multiple posts with their IDs, creation times, and messages. One message includes a link to a photo.

You can also access the number of likes and posts.

```
response = requests.get(url)

data = response.json()
#print(data)

posts = data['posts']
num_of_posts = len(posts['data'])
likes = data['likes']
#print(likes)
num_of_likes = len(likes['data'])
print(f"Number of Posts: {num_of_posts}") #TODO: numbers are not yet in total as we need to collect per paging request
print(f"Number of Likes: {num_of_likes}") #TODO: numbers are not yet in total as we need to collect per paging request
#print(posts['data'])

Number of Posts: 25
Number of Likes: 25
```

Task 4: Setting up a data streaming pipeline with Kafka to integrate data for data analysis

Kafka offers a base, for constructing data streaming systems that guarantee data accuracy and resilience crucial for tasks, like data analysis and real-time processing.

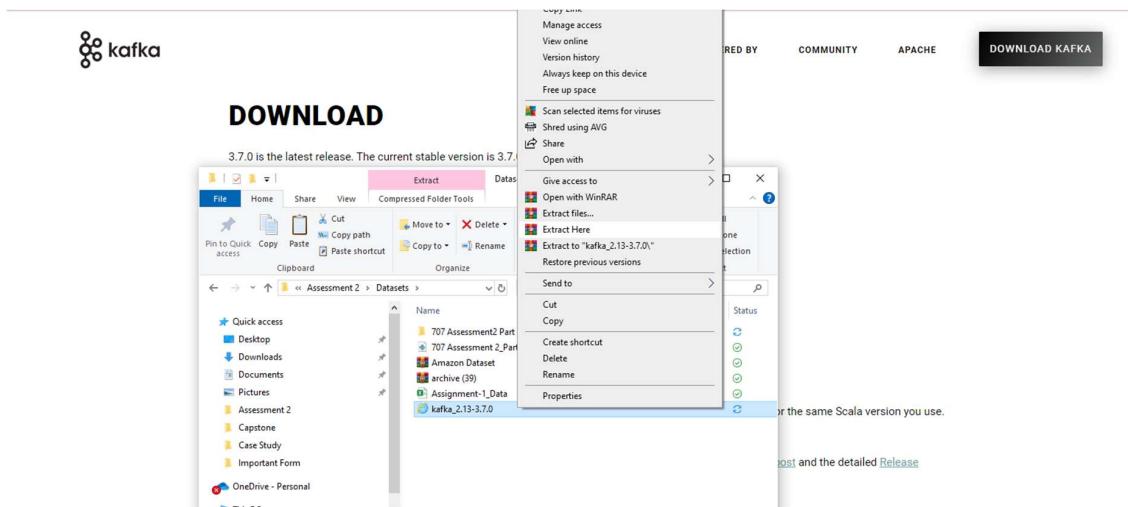
Steps on setting up a data streaming pipeline with kafka

1. Download the latest version of Kafka (kafka_2.13-3.7.0.tgz (asc, sha512))

The screenshot shows the Apache Kafka website at <https://kafka.apache.org/08/app-form.html>. The main navigation bar includes links for 'GET STARTED', 'DOCS', 'POWERED BY', 'COMMUNITY', 'APACHE', and a prominent 'DOWNLOAD KAFKA' button. Below the navigation, a large 'DOWNLOAD' section is visible, featuring a '3.7.0' heading and a bulleted list of download options. The list includes: 'Released Feb 27, 2024', 'Release Notes', 'Docker image: apache/kafka:3.7.0', 'Source download: kafka_3.7.0-src.tgz (asc, sha512)', and 'Binary downloads: Scala 2.12 - kafka_2.12-3.7.0.tgz (asc, sha512) and Scala 2.13 - kafka_2.13-3.7.0.tgz (asc, sha512)'. A note at the bottom states: 'We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).'

2. I used the WinRAR app to extract the files from kafka_2.13-3.7.0.tgz (zip file)

The screenshot shows the WinRAR website at <https://www.win-rar.com/postdownload.html?&L=0>. The top navigation bar includes links for 'SEARCH', 'HOME', 'ABOUT', 'NEWS', 'CONTACT', 'PRIVACY POLICY', and 'AGREE'. Below the navigation, the WinRAR logo is displayed. A message at the top says: 'Thank you for downloading WinRAR! If your download doesn't start within 5 seconds: [click here](#)'. A central box contains the text: 'The RAR Secrets Courses are now available online! Would you like to do more with WinRAR.exe and learn how to create your own RAR applications? Discover how to use RAR.exe, with its 100+ functions, with the new, updated RAR Command Line (RCL) course: all for just \$9.00! That's only \$0.75 per lesson! Join the thousands of users who already enrolled in 2023 [here!](#) Find out more...'. At the bottom of the page, there are links for 'PRODUCTS', 'DOWNLOAD', 'INDUSTRIES', 'PARTNER', 'SUPPORT', and 'NEWS', along with social media icons for Facebook, Twitter, and YouTube, and a 'PRIVACY | IMPRINT' link.



3. Move the extracted file to drive C to avoid command line errors (The input line is too long) shown below

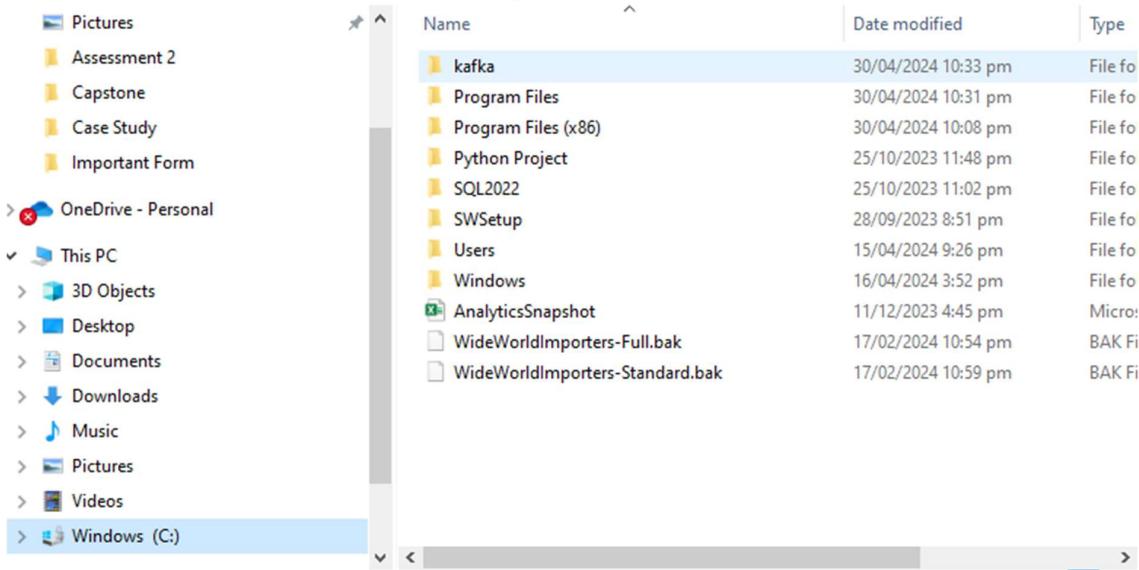
```
C:\Windows\System32\cmd.exe - .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\torri\OneDrive\Desktop\Mira\Data Analytics\707 Data Engineering\Assessment 2\Datasets\kafka_2.13-3.7.0>.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
The input line is too long.
The syntax of the command is incorrect.

C:\Users\torri\OneDrive\Desktop\Mira\Data Analytics\707 Data Engineering\Assessment 2\Datasets\kafka_2.13-3.7.0>
```

A screenshot of a Windows Command Prompt window titled 'cmd.exe'. The command entered is '.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties'. The output shows an error message: 'The input line is too long. The syntax of the command is incorrect.' The command prompt is located in the directory 'C:\Users\torri\OneDrive\Desktop\Mira\Data Analytics\707 Data Engineering\Assessment 2\Datasets\kafka_2.13-3.7.0'.

** Putting kafka in Drive C will simplify the command



4. Update the kafka server and zookeeper in the config file

** Updated kafka server

```

# The maximum size of a request that the socket server will accept (protection against OOM)
socket.request.max.bytes=104857600

# A comma-separated list of directories under which to store log files
log.dirs=/kafka/kafka-logs

# The default number of log partitions per topic. More partitions allow greater
# parallelism for consumption, but this will also result in more files across
# the cluster for a given topic.
num.partitions=1

# The number of threads per data directory to be used for log recovery at startup and flushing at shutdown.
# This value is recommended to be increased for installations with data dirs located in RAID array.
num.recovery.threads.per.data.dir=1

# Internal Topic Settings
# The replication factor for the group metadata internal topics "_consumer_offsets" and "_transaction_state"
# For anything other than development testing, a value greater than 1 is recommended to ensure availability such as 3.
# offset.storage=checkpointed
# transaction.state.log.replication.Factor=1
transaction.state.log.replication.Factor=1
transaction.state.log.min.isrs=1

# Log Flush Policy
# Messages are immediately written to the filesystem but by default we only sync() to sync
# the OS cache lazily. The following configurations control the flush of data to disk.

```

** Updated zookeeper

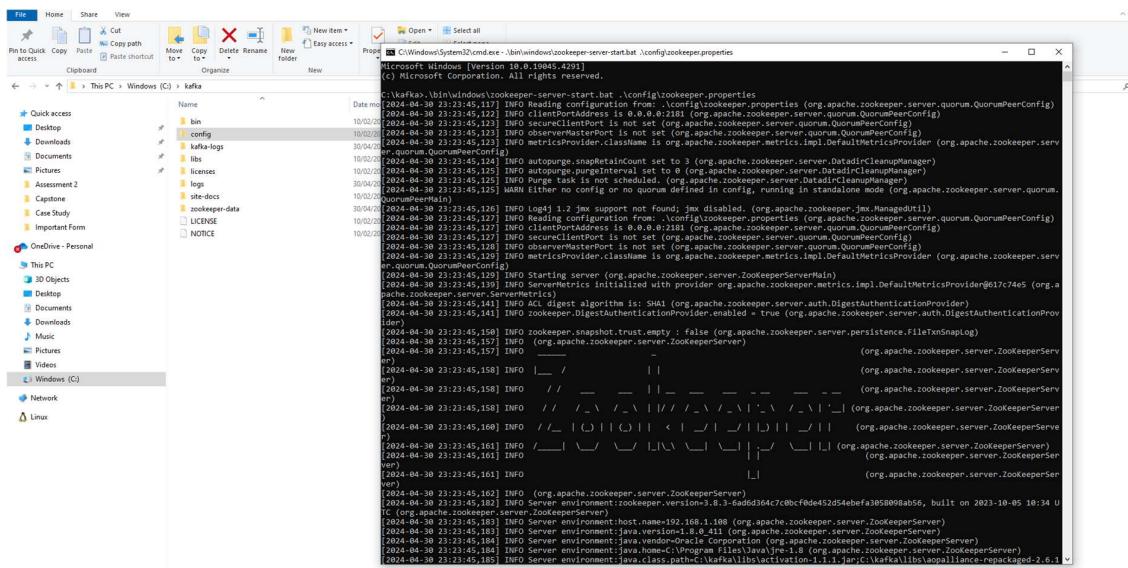
```

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to you under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

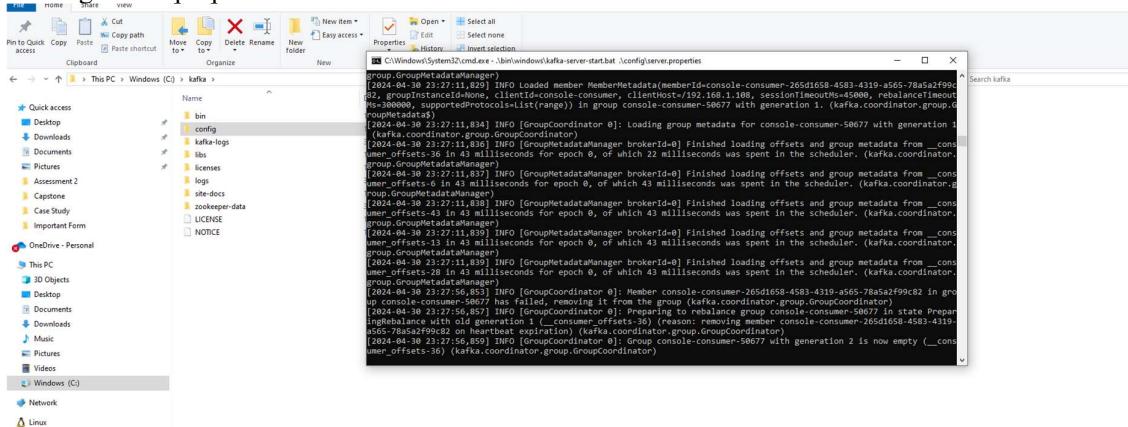
# the port at which the clients will connect
clientPort=2181
# allows per-ip limit on the number of connections since this is a non-production config
maxClientCnxns=0
# Disables the leader election
admin.enableServer=false
# admin.serverPort=9080

```

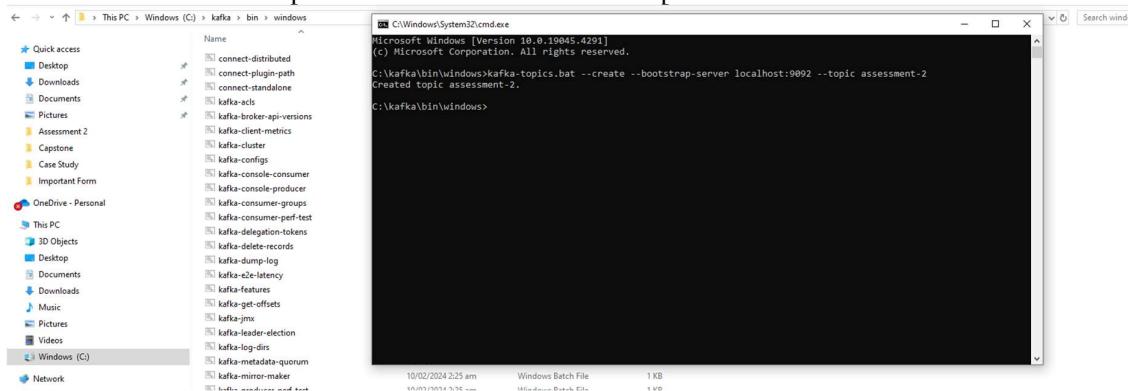
5. Run the zookeeper in the command line (cmd) using the code `.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties`



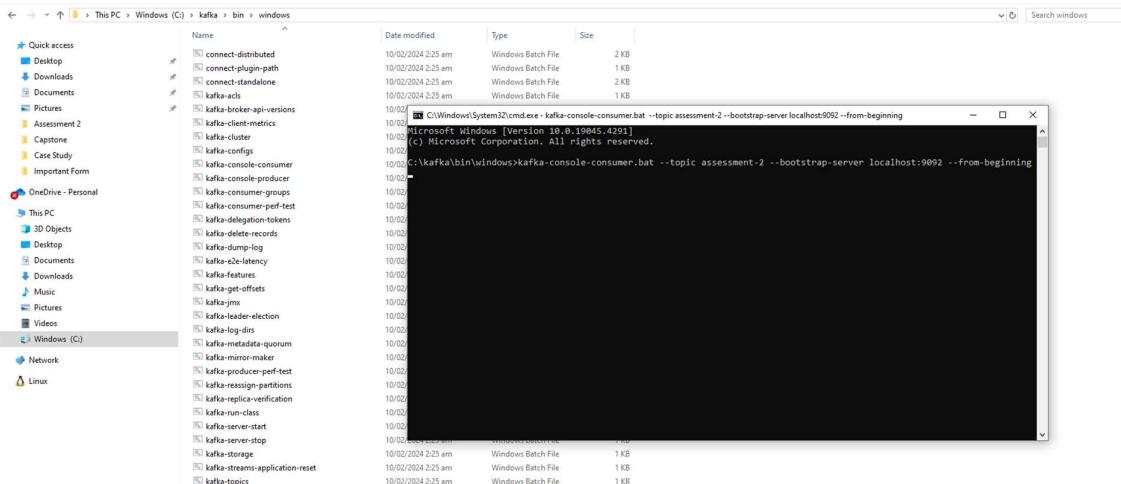
6. Run the kafka server in the command line(cmd) using the code `.\bin\windows\kafka-server-start.bat .\config\server.properties`



7. Create a kafka topic that serves as a table in kafka. Run the code `kafka-topics.bat --create --bootstrap-server localhost:9092 --topic assessment-2` in bin/windows path.



8. Run the consumer command in kafka which will receive the data



** Python code that will send data to kafka

```
In [ ]: from confluent_kafka import Producer

def delivery_report(err, msg):
    if err is not None:
        print(f"Message delivery failed: {err}")
    else:
        print(f"Message delivered to {msg.topic()} [{msg.partition()}] at offset {msg.offset()}")


p = Producer({'bootstrap.servers': "localhost:9092"})
p.produce("assessment-2", key="Assessment2", value=f"Part B , Task 4", callback=delivery_report)
p.flush()
```

** Sending data from Python to kafka

(Sample 1)

```
In [6]: from confluent_kafka import Producer

def delivery_report(err, msg):
    if err is not None:
        print(f"Message delivery failed: {err}")
    else:
        print(f"Message delivered to {msg.topic()} [{msg.partition()}] at offset {msg.offset()}")


p = Producer({'bootstrap.servers': "localhost:9092"})
p.produce("assessment-2", key="Assessment2", value=f"Part B , Task 4", callback=delivery_report)
p.flush()
```

Message delivered to assessment-2 [0] at offset 0

```
Out[6]: 0
In [ ]: C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --topic assessment-2 --bootstrap-server localhost:9092 --from-beginning
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\kafka\bin\windows>kafka-console-consumer.bat --topic assessment-2 --bootstrap-server localhost:9092 --from-beginning
Part B , Task 4
```

(Sample 2)

The screenshot shows a Jupyter Notebook interface. In the top navigation bar, it says "Jupyter Untitled20 Last Checkpoint: an hour ago (unsaved changes)". Below the menu bar, there are buttons for File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Run, Stop, Cell, Kernel, Widgets, Help, Logout, Trusted, Python 3 (ipykernel), and a dropdown menu.

In the main area, there is an "In [7]:" cell containing Python code for a Kafka producer:

```
from confluent_kafka import Producer

def delivery_report(err, msg):
    if err is not None:
        print(f"Message delivery failed: {err}")
    else:
        print(f"Message delivered to {msg.topic()} [{msg.partition()}] at offset {msg.offset()}")
p = Producer({'bootstrap.servers': "localhost:9092"})

for i in range(10):
    p.produce('assessment-2', key=f"Assessment-{i}", value=f" Mira {i}", callback=delivery_report)
p.flush()

Message delivered to assessment-2 [0] at offset 1
Message delivered to assessment-2 [0] at offset 2
Message delivered to assessment-2 [0] at offset 3
Message delivered to assessment-2 [0] at offset 4
Message delivered to assessment-2 [0] at offset 5
Message delivered to assessment-2 [0] at offset 6
Message delivered to assessment-2 [0] at offset 7
Message delivered to assessment-2 [0] at offset 8
Message delivered to assessment-2 [0] at offset 9
Message delivered to assessment-2 [0] at offset 10
```

Below the code, the "Out[7]:" cell shows the output of the producer's delivery report:

```
Out[7]: 0
```

A separate window titled "C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --topic assessment-2 --bootstrap-server localhost:9092 --from-beginning" is running in the background, displaying the received messages:

```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\kafka\bin\windows>kafka-console-consumer.bat --topic assessment-2 --bootstrap-server localhost:9092 --from-beginning
Part B , Task 4
Mira 0
Mira 1
Mira 2
Mira 3
Mira 4
Mira 5
Mira 6
Mira 7
Mira 8
Mira 9
```

** Sending data from kafka to Python – run the code kafka-console-producer.bat --broker-list localhost:9092 --topic assessment-2-b-4 to allow sending data from kafka

The screenshot shows a Windows command prompt window. It displays the following commands and their outputs:

```
C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic assessment-2-b-4
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\kafka\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --topic assessment-2-b-4
Created topic assessment-2-b-4.
```

** Python code to retrieve data from kafka

The screenshot shows a Jupyter Notebook interface. In the top navigation bar, it says "Jupyter Untitled20 Last Checkpoint: a few seconds ago (unsaved changes)". Below the menu bar, there are buttons for File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Run, Stop, Cell, Kernel, Widgets, Help, Logout, Trusted, Python 3 (ipykernel), and a dropdown menu.

In the main area, there is an "In []:" cell containing Python code for a Kafka consumer:

```
from confluent_kafka import Consumer, KafkaError

c = Consumer({'bootstrap.servers': "localhost:9092", "group.id": "group_kafka_python1"})
c.subscribe(['assessment-2-b-4'])

while True:
    msg = c.poll(1.0)
    if msg is None:
        continue
    if msg.error():
        if msg.error().code() == KafkaError._PARTITION_EOF:
            print(f'Reached end of partition {msg.partition()}')
        else:
            print(f'Error: {msg.error()}')
    else:
        print(f'Received message: {msg.value().decode("utf-8")}')
c.close()
```

The screenshot shows a Jupyter Notebook interface with a Python 3 (ipykernel) kernel. In the code cell (In [1]), the following Python code is displayed:

```
from confluent_kafka import Consumer, KafkaError
c = Consumer({'bootstrap.servers': "localhost:9092", "group.id": "group_kafka_python1"})
c.subscribe(['assessment-2-b-4'])

while True:
    msg = c.poll(1.0)
    if msg is None:
        continue
    if msg.error():
        if msg.error().code() == KafkaError._PARTITION_EOF:
            print(f'Reached end of partition {msg.partition()}')
        else:
            print(f'Error: {msg.error()}')
    else:
        print(f'Received message: {msg.value().decode("utf-8")}')
c.close()

Received message: This message is from kafka
Received message: This message is from kafka
```

Below the code cell is an output cell (In [1]) which shows the terminal command and its output:

```
C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic assessment-2-b-4
Microsoft Windows [Version 10.0.19045.429]
(c) Microsoft Corporation. All rights reserved.

C:\kafka\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --topic assessment-2-b-4
Created topic assessment-2-b-4.

C:\kafka\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic assessment-2-b-4
>This message is from kafka
>This message is from kafka
>
```

Conclusion:

Data integration is essential in business, especially with big data. However, it can be a very risky process if not done correctly. Most errors occur in connection and configuration, thus requiring attention to detail. Knowledge of the different integration processes will also help manipulate and treat the data. Data cleaning is always the first step to ensure the quality of the data. As per experience in the tasks, integration planning is crucial to facilitate the integration and avoid errors during the process. Familiarization with the data structure is also needed to choose the proper storage or database.

References

Ahmedov, A. (n.d.). Market Basket Analysis: Analyzing Consumer Behaviour Using MBA Association Rule Mining. Kaggle.

<https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis/data>

Anil. (n.d.). data.world. <https://data.world/anilsharma87>

E-Commerce Data. (n.d.). Customer Segmentation & Recommendation. Kaggle.

<https://www.kaggle.com/code/farzadnekouei/customer-segmentation-recommendation-system/input>

The Devastator.(n.d.). E-Commerce Sales Dataset.Kaggle.

<https://www.kaggle.com/datasets/thedevastator/unlock-profits-with-e-commerce-sales-data>

Ramos, G. (n.d.). E-commerce Business Transaction: Sales transaction of a UK-based e-commerce

(online retail) for one year. Kaggle.

<https://www.kaggle.com/datasets/gabrielramos87/an-online-shop-business/data>