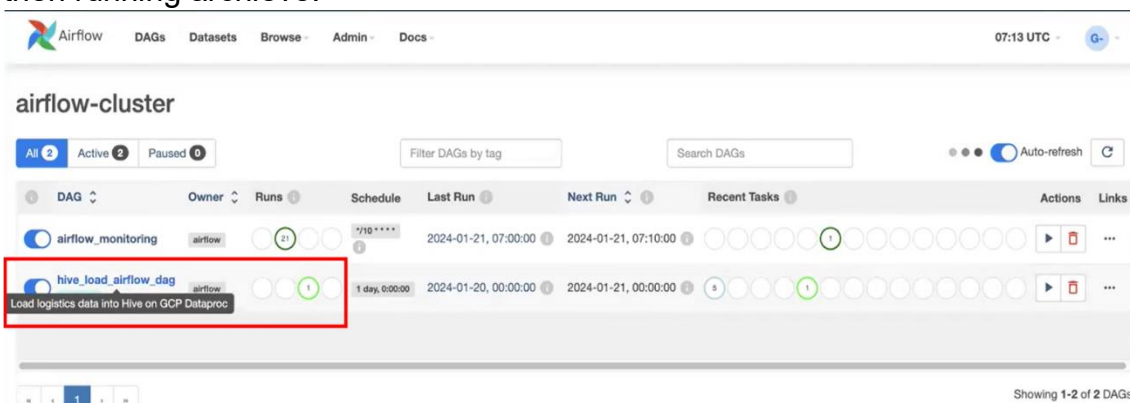
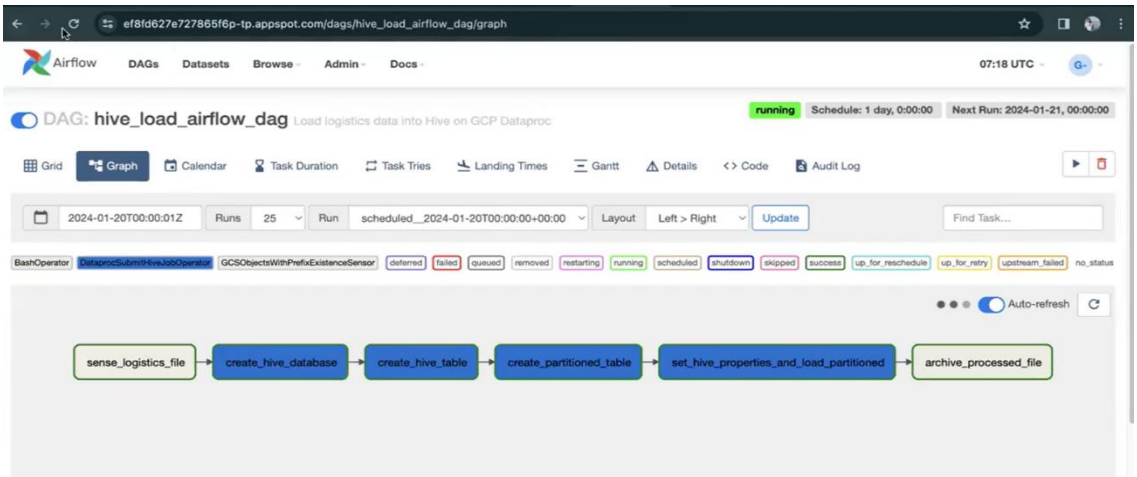
Logistics Data Warehouse Management

1. we had a gcs bucket named 'logistic-raw' where we were receiving daily csv file with file name having date in it.

	A	B	C	D	E	F	G	H
1	delivery_i	date	origin	destination	vehicle_ty	delivery_sta	delivery_time	
2	26	02-09-2023	New York	Boston	Bike	Delivered	7 hours	
3	27	02-09-2023	Washington	Philadelphia	Truck	In-Transit	5 hours	
4	28	02-09-2023	San Francisco	San Diego	Drone	Delivered	2.5 hours	
5	29	02-09-2023	Austin	Houston	Truck	Cancelled	4 hours	
6	30	02-09-2023	Orlando	Miami	Truck	Delivered	4.5 hours	
7	31	02-09-2023	Detroit	Chicago	Drone	Delivered	3 hours	
8	32	02-09-2023	Portland	Seattle	Bike	In-Transit	2.5 hours	
9	33	02-09-2023	Phoenix	Las Vegas	Truck	Delivered	6 hours	
10	34	02-09-2023	Philadelphia	New York	Drone	Delivered	2 hours	
11	35	02-09-2023	Memphis	Nashville	Bike	Cancelled	2.5 hours	
12	36	02-09-2023	Los Angeles	San Francisco	Truck	Delivered	5 hours	
13	37	02-09-2023	Charlotte	Atlanta	Bike	In-Transit	3.5 hours	

2. next set up an airflow job used to listen the file received in bucket. We perfomed certain stages in airflow job which includes create database, create external stage table, create partition table, loading data into partition table and then running archieve.





3. In airflow dag job, we were using gcs cloud storage sensor here to sense the file landing in the gcs storage bucket and DataprocSubmitHiveJobOperator to run, submit hive queries on the dataproc cluster.

```
dag = DAG(
    'hive_load_airflow_dag',
    default_args=default_args,
    description='Load logistics data into Hive on GCP Dataproc',
    schedule_interval=timedelta(days=1),
    start_date=days_ago(1),
    tags=['example'],
)

# Sense the new file in GCS
sense_logistics_file = GCSObjectsWithPrefixExistenceSensor(
    task_id='sense_logistics_file',
    bucket='logistics-raw-gds',
    prefix='input_data/logistics_',
    mode='poke',
    timeout=300,
    poke_interval=30,
    dag=dag
)


# Create Hive Database if not exists
create_hive_database = DataprocSubmitHiveJobOperator(
    task_id="create_hive_database",
    query="CREATE DATABASE IF NOT EXISTS logistics_db;",
    cluster_name='compute-cluster',
    region='us-central1',
    project_id='big-data-projects-411817',
    dag=dag
)
```

4. We created external stage table on the 'logistic-raw' bucket path. We ensured that only delta data or current data should be present in this path. External table was being used in order to ingest the data in partitioned form into the hive partition table.

```
# Create main Hive table
create_hive_table = DataprocSubmitHiveJobOperator(
    task_id="create_hive_table",
    query="""
        CREATE EXTERNAL TABLE IF NOT EXISTS logistics_db.logistics_data (
            delivery_id INT,
            `date` STRING,
            origin STRING,
            destination STRING,
            vehicle_type STRING,
            delivery_status STRING,
            delivery_time STRING
        )
        ROW FORMAT DELIMITED
        FIELDS TERMINATED BY ','
        STORED AS TEXTFILE
        LOCATION 'gs://logistics-raw-gds/input_data/'
        tblproperties('skip.header.line.count'='1');
    """,
    cluster_name='compute-cluster',
    region='us-central1',
    project_id='big-data-projects-411817',
    dag=dag
)
```

```python
# Create partitioned Hive table
create_partitioned_table = DataprocSubmitHiveJobOperator(
    task_id="create_partitioned_table",
    query="""
        CREATE TABLE IF NOT EXISTS logistics_db.logistics_data_partitioned (
            delivery_id INT,
            origin STRING,
            destination STRING,
            vehicle_type STRING,
            delivery_status STRING,
            delivery_time STRING
        )
        PARTITIONED BY (`date` STRING)
        STORED AS TEXTFILE;
    """,
    cluster_name='compute-cluster',
    region='us-central1',
    project_id='big-data-projects-411817',
    dag=dag
)
```

```python
# Set Hive properties for dynamic partitioning and load data
set_hive_properties_and_load_partitioned = DataprocSubmitHiveJobOperator(
    task_id="set_hive_properties_and_load_partitioned",
    query=f"""
        SET hive.exec.dynamic.partition = true;
        SET hive.exec.dynamic.partition.mode = nonstrict;

        INSERT INTO logistics_db.logistics_data_partitioned PARTITION(`date`)
        SELECT delivery_id, origin,destination, vehicle_type, delivery_status,
        delivery_time, `date` FROM logistics_db.logistics_data;
    """,
    cluster_name='compute-cluster',
    region='us-central1',
    project_id='big-data-projects-411817',
    dag=dag
)
```

5. We have another bucket 'logistics-archieve' where we used to move daily file from 'logistic-raw' after file is processed. In this way we archieved processed file using the BashOperator.

```python
# Move processed files to archive bucket
archive_processed_file = BashOperator(
    task_id='archive_processed_file',
    bash_command=f"gsutil -m mv
    gs://logistics-raw-gds/input_data/logistics_*.csv
    gs://logistics-archive-gds/",
    dag=dag
)
```

6. Here, the query results for hive partitioned table

ssh.cloud.google.com/v2/ssh/projects/big-data-projects-411817/zones/us-central1-a/instances/compute-cluster-m?authuser=1&hl=en_US&projectNumber=3...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.hive.common.StringInternUtils (file:/usr/lib/hive/
on-3.1.3.jar) to field java.net.URI.string
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.hive.common.StringInternUt
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Hive Session ID = 4a44523e-9234-48bf-a559-ee267b4720a5
hive> show databases;
OK
default
logistics_db
Time taken: 1.066 seconds, Fetched: 2 row(s)
hive> use logistics_db;
OK
Time taken: 0.085 seconds
hive> show tables;
OK
logistics_data
logistics_data_partitioned
Time taken: 0.08 seconds, Fetched: 2 row(s)
hive> select * from logistics_data;
OK
Time taken: 2.333 seconds
hive> select * from logistics_data_partitioned limit 5;
OK
1       New York        New Jersey      Truck   Delivered       5 hours 2023-09-01
2       Boston  Washington      Truck   In-Transit      8 hours 2023-09-01
3       Los Angeles     San Francisco   Drone   Delivered       2 hours 2023-09-01
4       Dallas  Austin  Bike    Cancelled       3 hours 2023-09-01
5       Miami   Orlando Truck   Delivered       4 hours 2023-09-01
Time taken: 0.572 seconds, Fetched: 5 row(s)
hive>
```