

Name: Abhishek Obla Hema
Email: oh_abhishek@yahoo.com

Airflow Assignment-1 Documentation

This file details and describes all the attached files for the Airflow Assignment -1

Tools Used:

1. Python3 – Microsoft VScode
2. Apache Airflow
3. GCP services – DataProc/GCS
4. Apache Hive

Files Attached:

1. Airflow_assignment_1.pdf – This file
2. Employee_batch.py – Pyspark job that filters employees with salary $\geq 60,000$ and stores it in a GCS location.
3. Airflow_ass1_job.py – Airflow job that details the DAGs
4. Employee.csv – Csv file used for the exercise

Process and File Descriptions:

Step 1:

I created a bucket called 'airflow_ass1' and placed employee.csv under input_files folder. This file will be picked up by the spark job which is a part of the DAG

airflow_ass1

Location

Storage class

Public access

Protection

us (multiple regions in United States)

Standard

Not public

None

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

Buckets

>

airflow_ass1

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS

DOWNLOAD




DELETE

Filter by name prefix only

Filter

Filter objects and folders

Show deleted data

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption
<input type="checkbox"/>	 hive_data/	—	Folder	—	—	—	—	—	—
<input type="checkbox"/>	 input_files/	—	Folder	—	—	—	—	—	—
<input type="checkbox"/>	 python_file/	—	Folder	—	—	—	—	—	—

I also made sure to place the pyspark job 'employee_batch.py' in the python_file folder in the same GCS bucket.

Step 2:

I create an airflow cluster and then proceeded to place the ‘airflow_ass1_job.py’ file in the DAG list so that it can be picked up by airflow

Google Cloudtest-projectcloud composerSearch

ComposerEnvironmentsCREATEREFRESHDELETE

Filter environments

State	Name	Location	Composer version	Airflow version	Creation time	Update time	Airflow webserver	DAG list	Logs	DAGs folder	Labels
<input checked="" type="checkbox"/>	airflow-cluster	us-central1	1.20.12	2.4.3	31/12/2023, 21:11	31/12/2023, 21:27	Airflow	DAGs	Logs	DAGs	None

us-central1-airflow-cluster-e06b719c-bucket

LocationStorage classPublic accessProtection

us-central1 (Iowa)StandardSubject to object ACLsNone

OBJECTS

CONFIGURATION

PERMISSION

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

Buckets > us-central1-airflow-cluster-e06b719c-bucket > dags

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS

DOWNLOAD

DELETE

Filter by name prefix onlyFilter objects and foldersShow deleted data

Name	Size	Type	Created	Storage class	Last modified	Public access
airflow_ass1_job.py	2.7 KB	text/x-python-script	31 Dec 2023, 21:58:49	Standard	31 Dec 2023, 21:58:49	Not public
airflow_ass2_job.py	2.2 KB	text/x-python-script	1 Jan 2024, 06:31:12	Standard	1 Jan 2024, 06:31:12	Not public
airflow_monitoring.py	809 B	text/x-python	31 Dec 2023, 21:24:55	Standard	31 Dec 2023, 21:24:55	Not public

Step 3:

Now we can see the various stages in the DAG. A file sensor checks every 5 mins in the input_file location, once it detects employee.csv a new cluster is created followed by which the spark job is launched which then filters employees with salary >=60,000 and then places the output in another GCS location in the bucket airflow_ass1 under the hive_data folder.

DAG: gcp_dataproc_spark_jobA DAG to run Spark job on DataprocSchedule: 1 day, 0:00:00Next Run: 2024-01-01, 00:00:00

GridGraphCalendarTask DurationTask TriesLanding TimesGanttDetailsCodeAudit Log

01/01/2024, 01:49:17 PM25All Run TypesAll Run StatesClear Filters

deferredfailedqueuedremovedrestartingrunningscheduledshutdownskippedsuccessup_for_rescheduleup_for_retryupstream_failedno_status

Auto-refresh

Duration

00:16:46

00:08:23

00:00:00

Jan 01, 08:30

file_sensor_task

create_cluster

submit_pyspark_job

delete_cluster

DAG

gcp_dataproc_spark_job

DAG Details

DAG Runs Summary

Total Runs Displayed

10

Total success

8

Total failed

1

Total running

1

First Run Start

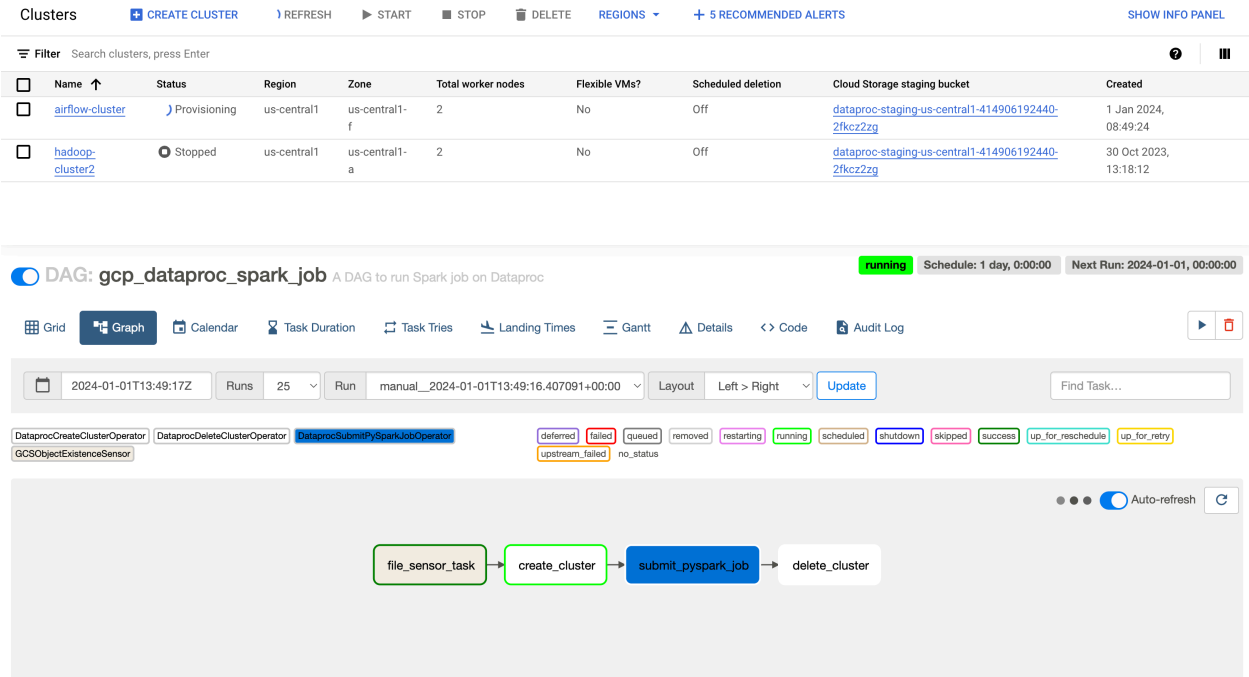
2024-01-01, 02:59:47 UTC

Last Run Start

2024-01-01, 13:49:17 UTC

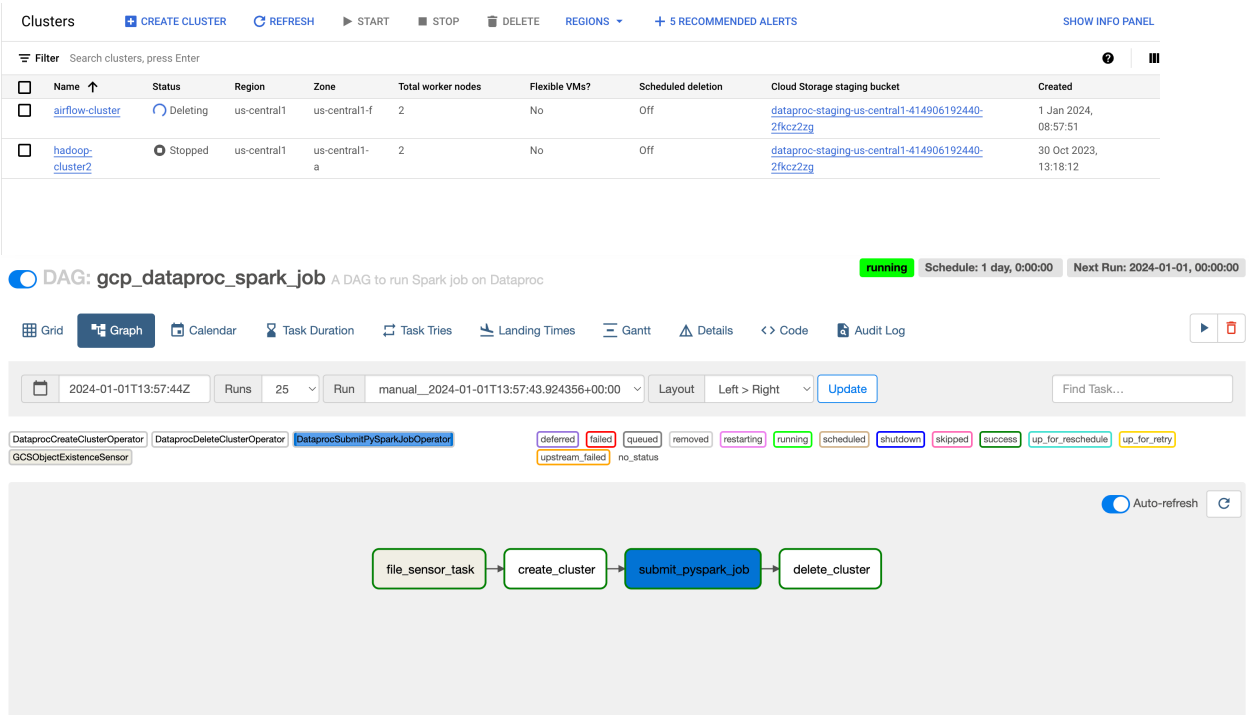
Max Run Duration

00:16:46



Step 4:

We can see that all stages have run successfully followed by deletion of the cluster as well



Step 5:

We can see the resultant data saved in a parquet file format saved in hive_data below (over which we can build external hive tables to query the data)

airflow_ass1

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE OBSERVABILITY INVENTORY REPORTS

Buckets > airflow_ass1 > hive_data > airflow.db > filtered_employee 📄

[UPLOAD FILES](#) [UPLOAD FOLDER](#) [CREATE FOLDER](#) [TRANSFER DATA](#) ▾ [MANAGE HOLDS](#) [DOWNLOAD](#) [DELETE](#)

Filter by name prefix only ▾ **Filter** Filter objects and folders Show deleted data

<input type="checkbox"/>	Name	Size	Type	Created ?	Storage class	Last modified	Public access	
<input type="checkbox"/>	part-00000-1903c7b0-87b3-43e2-...	861 B	application/octet-stream	Jan 1, 2024, 2:41:29 PM	Standard	Jan 1, 2024, 2:41:29 PM	Not public	

Challenges

1. Issues with the spark job being picked up since we had to define a location to save the output.
2. We can't put the output on the local of the cluster since with the deletion phase of the cluster this data would disappear as well. Hence it makes sense to place the output in a GCS bucket
3. Make sure to specify the format like "hive"/"parquet" while saving the output.