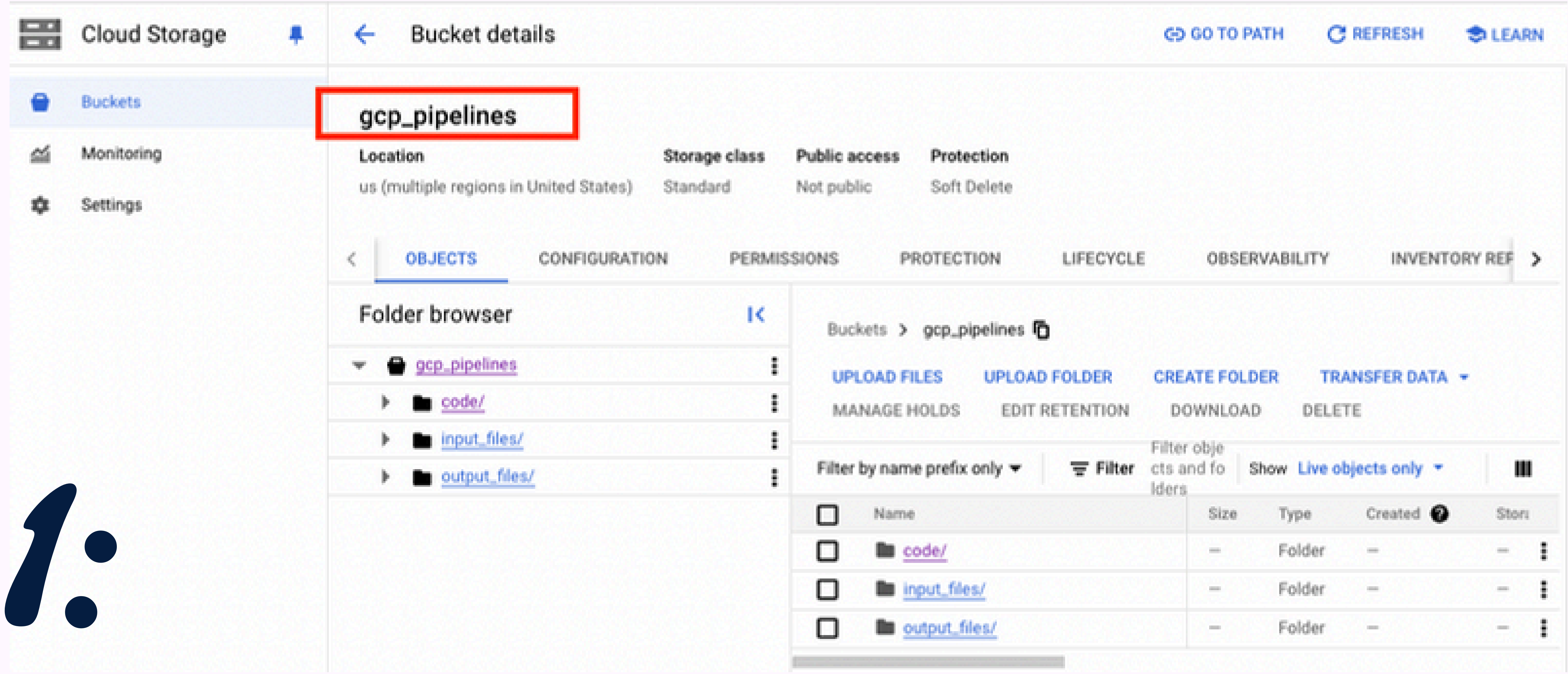
ETL GCP-Snowflakes

Then Tsze Yen



gcp_pipelines

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	Soft Delete

gcp_pipelines
- code/
- input_files/
- output_files/

Buckets > gcp_pipelines

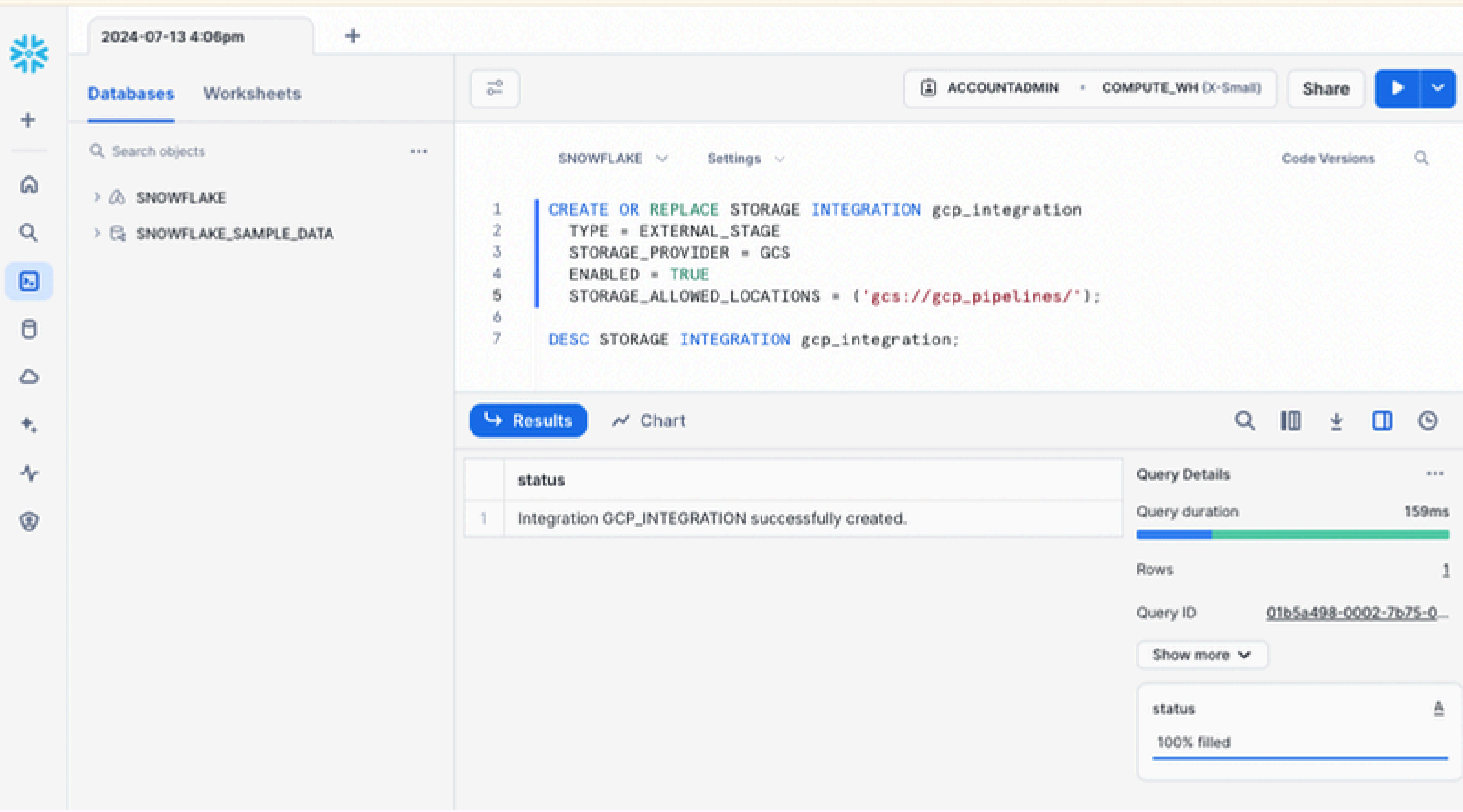
□	Name	Size	Type	Created	Stor
□	code/	—	Folder	—	—
□	input_files/	—	Folder	—	—
□	output_files/	—	Folder	—	—

Step 1:

Create Cloud Storage Bucket (to connect snowflakes, as my snowflakes is in us central region, then create bucket in us region)

Step 2 :

Create a Cloud Storage Integration in Snowflake



```sql
CREATE OR REPLACE STORAGE INTEGRATION gcp_integration
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = GCS
  ENABLED = TRUE
  STORAGE_ALLOWED_LOCATIONS = ('gcs://gcp_pipelines/');

DESC STORAGE INTEGRATION gcp_integration;
```

	status
1	Integration GCP_INTEGRATION successfully created.

Query Details

Query duration 159ms

Rows 1

Query ID 01b5a498-0002-7b75-0...

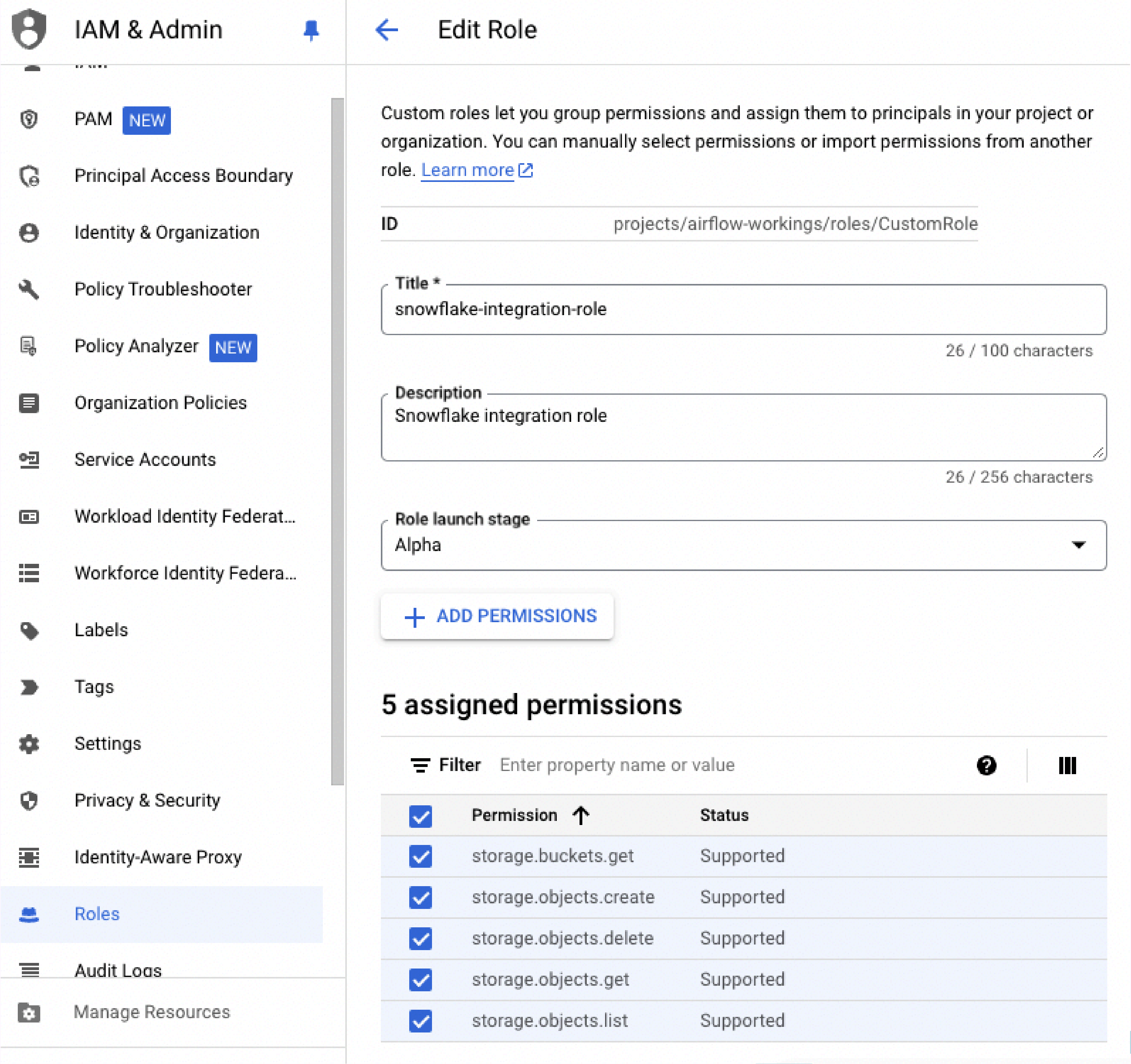
Show more

status

100% filled

Step 3:

Create Role



Custom roles let you group permissions and assign them to principals in your project or organization. You can manually select permissions or import permissions from another role. Learn more

ID projects/airflow-workings/roles/CustomRole

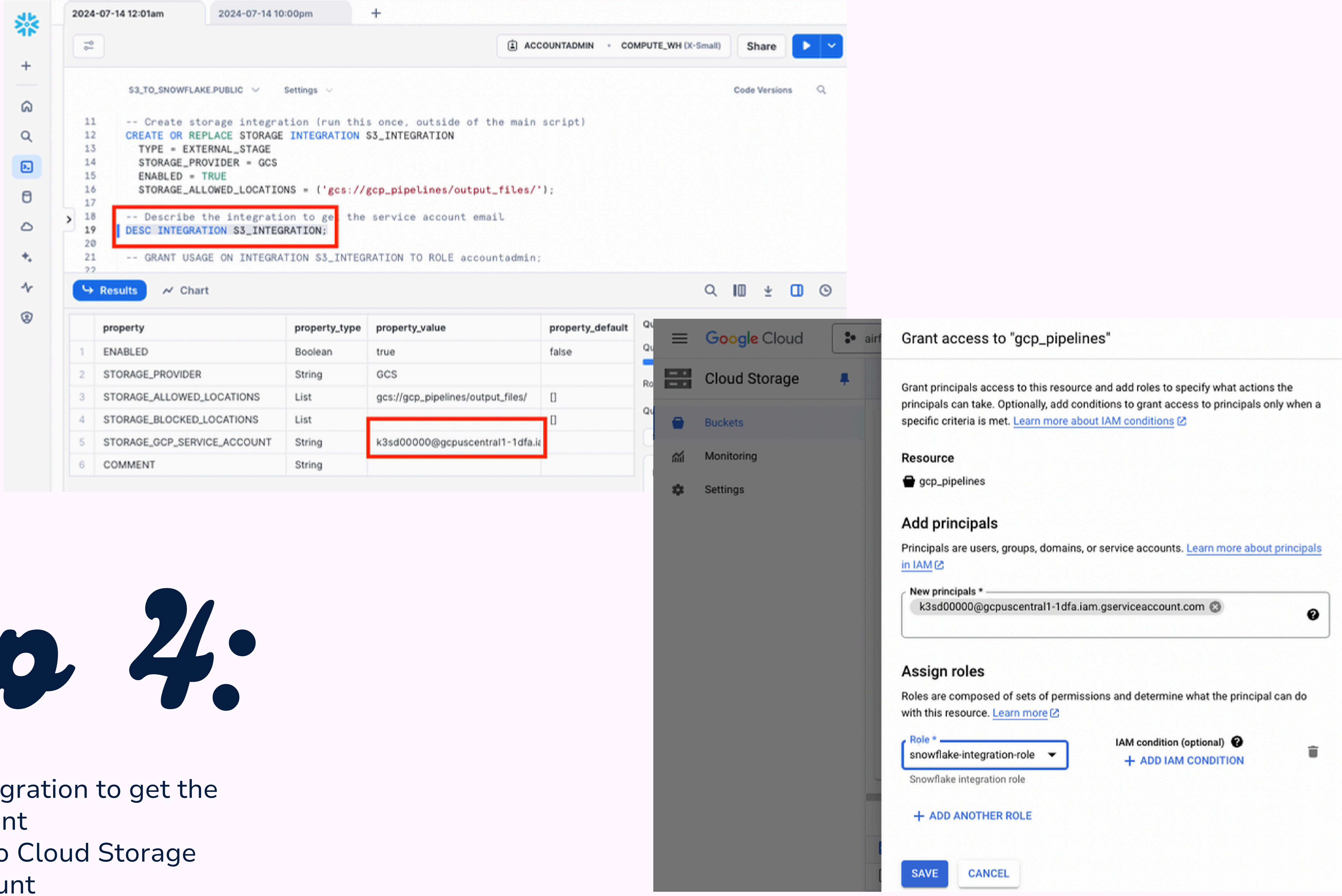
Title * snowflake-integration-role
26 / 100 characters

Description Snowflake integration role
26 / 256 characters

Role launch stage Alpha

5 assigned permissions

☑	Permission ↑	Status
☑	storage.buckets.get	Supported
☑	storage.objects.create	Supported
☑	storage.objects.delete	Supported
☑	storage.objects.get	Supported
☑	storage.objects.list	Supported



```
-- Create storage integration (run this once, outside of the main script)
CREATE OR REPLACE STORAGE INTEGRATION S3_INTEGRATION
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = GCS
  ENABLED = TRUE
  STORAGE_ALLOWED_LOCATIONS = ('gcs://gcp_pipelines/output_files/');

-- Describe the integration to get the service account email
DESC INTEGRATION S3_INTEGRATION;

-- GRANT USAGE ON INTEGRATION S3_INTEGRATION TO ROLE accountadmin;
```

	property	property_type	property_value	property_default
1	ENABLED	Boolean	true	false
2	STORAGE_PROVIDER	String	GCS	
3	STORAGE_ALLOWED_LOCATIONS	List	gcs://gcp_pipelines/output_files/	[]
4	STORAGE_BLOCKED_LOCATIONS	List		[]
5	STORAGE_GCP_SERVICE_ACCOUNT	String	k3sd00000@gcpuscentral1-1dfa.ia	
6	COMMENT	String		

Grant access to "gcp_pipelines"

Grant principals access to this resource and add roles to specify what actions the principals can take. Optionally, add conditions to grant access to principals only when a specific criteria is met. Learn more about IAM conditions

Resource

gcp_pipelines

Add principals

Principals are users, groups, domains, or service accounts. Learn more about principals in IAM

New principals *: k3sd00000@gcpuscentral1-1dfa.iam.gserviceaccount.com

Assign roles

Roles are composed of sets of permissions and determine what the principal can do with this resource. Learn more

Role *: snowflake-integration-role

Snowflake integration role

IAM condition (optional)

+ ADD IAM CONDITION

+ ADD ANOTHER ROLE

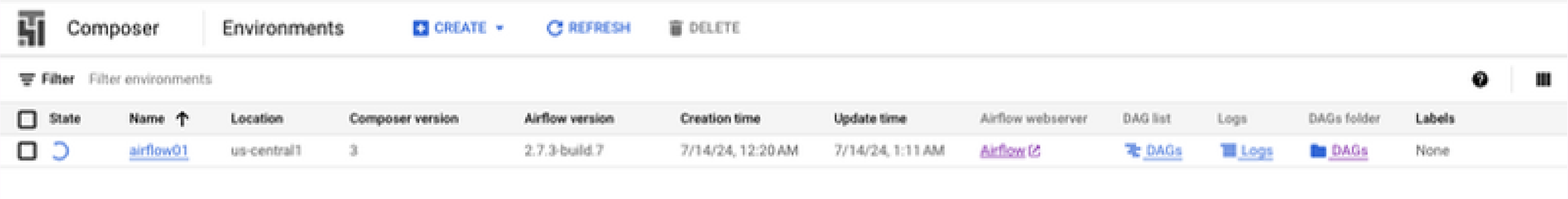
SAVE CANCEL

Step 4:

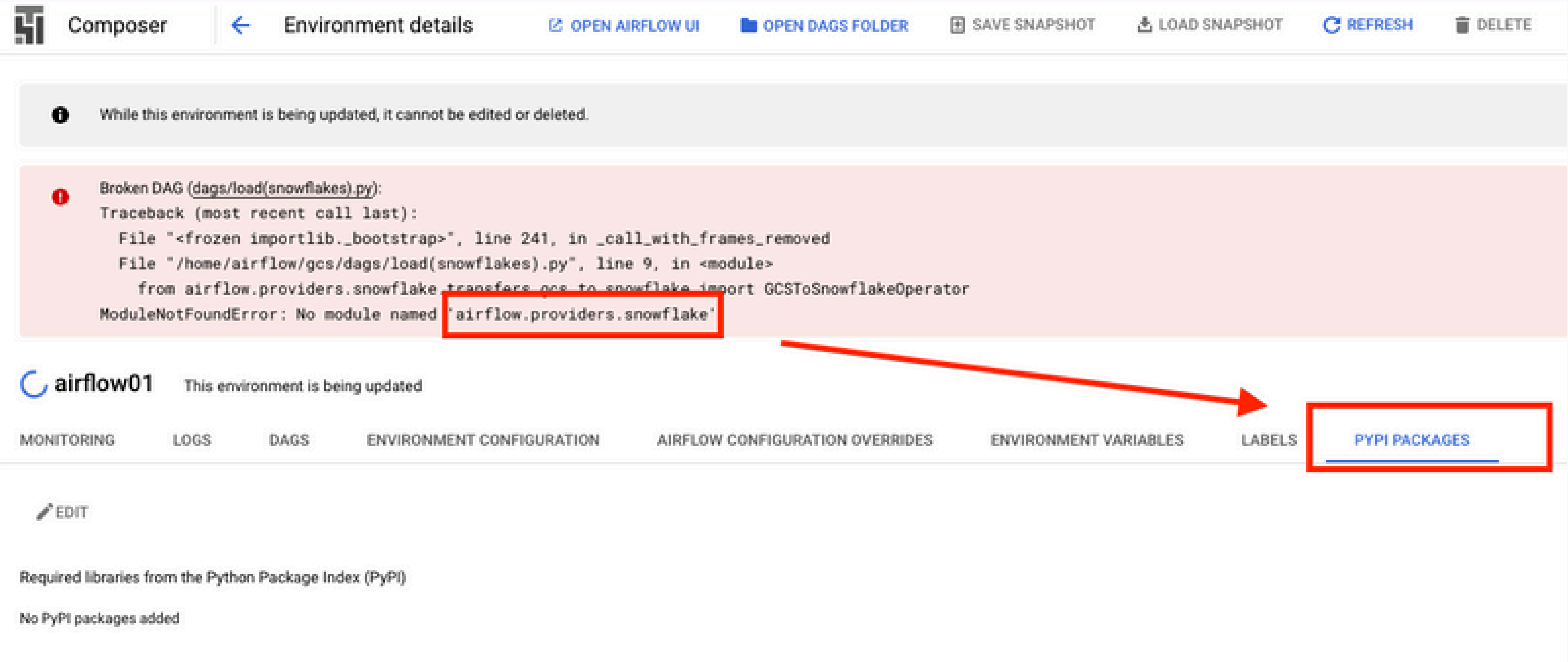
- Desc the integration to get the service account
- Assign role to Cloud Storage Service Account

Step 5:

Create Cloud Composer Environment & Add PYPI package



State	Name ↑	Location	Composer version	Airflow version	Creation time	Update time	Airflow webserver	DAG list	Logs	DAGs folder	Labels
□	airflow01	us-central1	3	2.7.3-build.7	7/14/24, 12:20 AM	7/14/24, 1:11 AM	Airflow	DAGs	Logs	DAGs	None



While this environment is being updated, it cannot be edited or deleted.

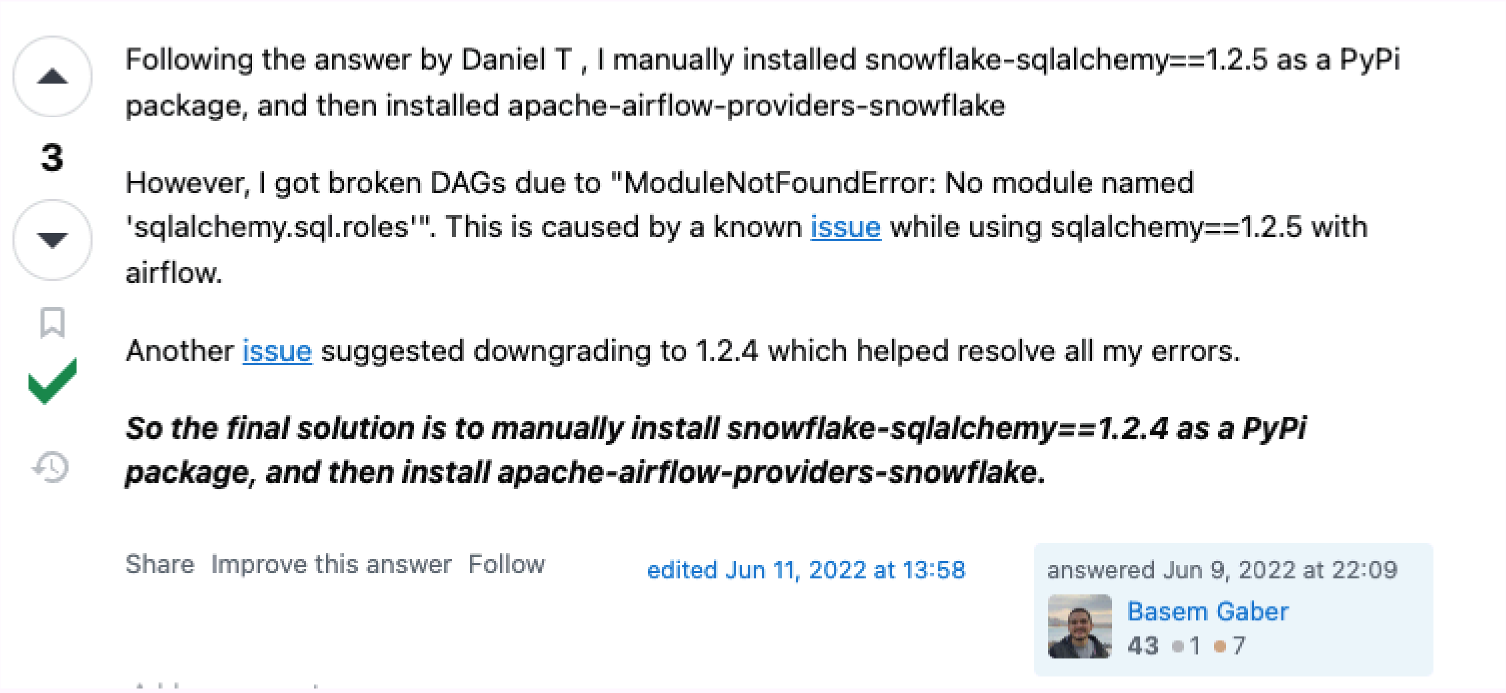
Broken DAG (dags/load(snowflakes).py):

```
Traceback (most recent call last):
  File "<frozen importlib._bootstrap>", line 241, in _call_with_frames_removed
  File "/home/airflow/gcs/dags/load(snowflakes).py", line 9, in <module>
    from airflow.providers.snowflake.transfers.gcs_to_snowflake import GCSToSnowflakeOperator
ModuleNotFoundError: No module named 'airflow.providers.snowflake'
```

airflow01 This environment is being updated

Required libraries from the Python Package Index (PyPI)

No PyPI packages added



Following the answer by Daniel T , I manually installed snowflake-sqlalchemy==1.2.5 as a PyPi package, and then installed apache-airflow-providers-snowflake

However, I got broken DAGs due to "ModuleNotFoundError: No module named 'sqlalchemy.sql.roles'". This is caused by a known issue while using sqlalchemy==1.2.5 with airflow.

Another issue suggested downgrading to 1.2.4 which helped resolve all my errors.

So the final solution is to manually install snowflake-sqlalchemy==1.2.4 as a PyPi package, and then install apache-airflow-providers-snowflake.

Share Improve this answer Follow

edited Jun 11, 2022 at 13:58

answered Jun 9, 2022 at 22:09

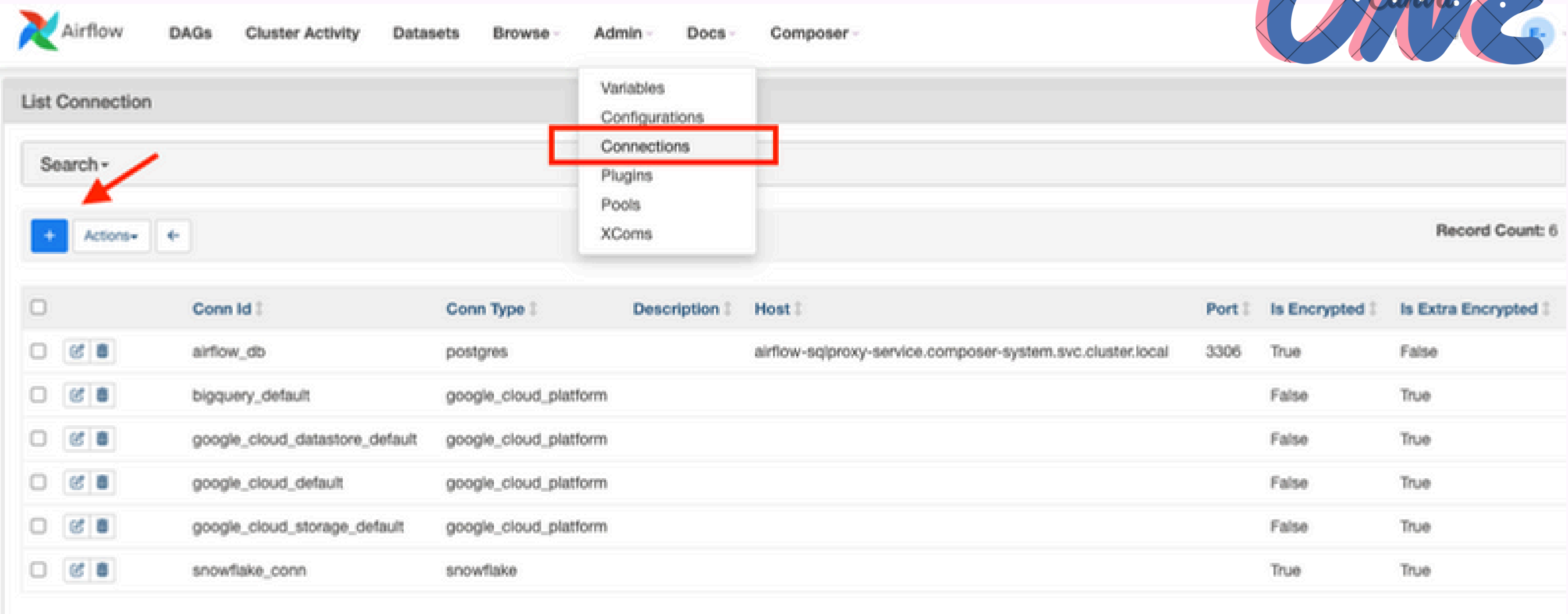
Basem Gaber

43 ●1 ●7

Step 6:

Add airflow-snowflake connection

ONE

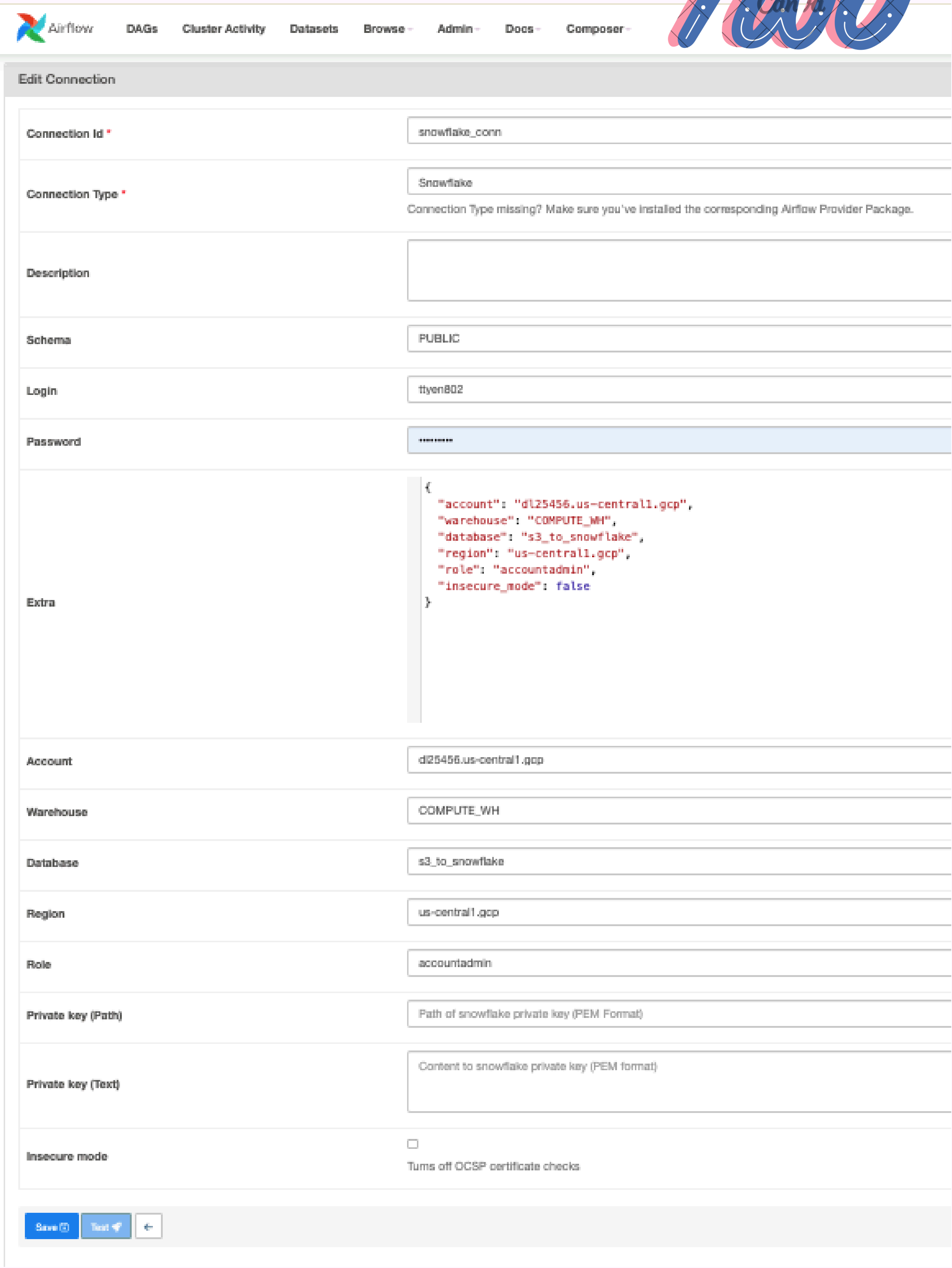


List Connection

Record Count: 6

☐		Conn Id	Conn Type	Description	Host	Port	Is Encrypted	Is Extra Encrypted
☐		airflow_db	postgres		airflow-sqlproxy-service.composer-system.svc.cluster.local	3306	True	False
☐		bigquery_default	google_cloud_platform				False	True
☐		google_cloud_datastore_default	google_cloud_platform				False	True
☐		google_cloud_default	google_cloud_platform				False	True
☐		google_cloud_storage_default	google_cloud_platform				False	True
☐		snowflake_conn	snowflake				True	True

TWO



Edit Connection

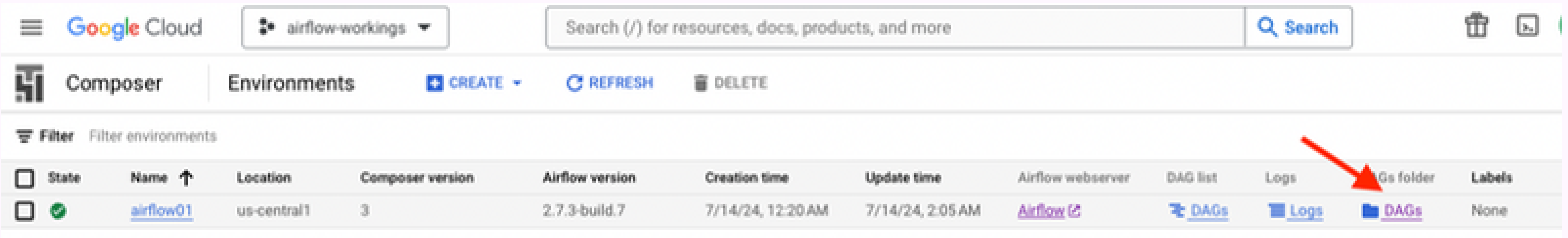
Connection Id *	snowflake_conn
Connection Type *	Snowflake Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.
Description	
Schema	PUBLIC
Login	ttyen802
Password	••••••••••
Extra	see below
Account	dl25456.us-central1.gcp
Warehouse	COMPUTE_WH
Database	s3_to_snowflake
Region	us-central1.gcp
Role	accountadmin
Private key (Path)	Path of snowflake private key (PEM Format)
Private key (Text)	Content to snowflake private key (PEM format)
Insecure mode	☐ Turns off OCSP certificate checks

```
{
  "account": "dl25456.us-central1.gcp",
  "warehouse": "COMPUTE_WH",
  "database": "s3_to_snowflake",
  "region": "us-central1.gcp",
  "role": "accountadmin",
  "insecure_mode": false
}
```

Step 6:

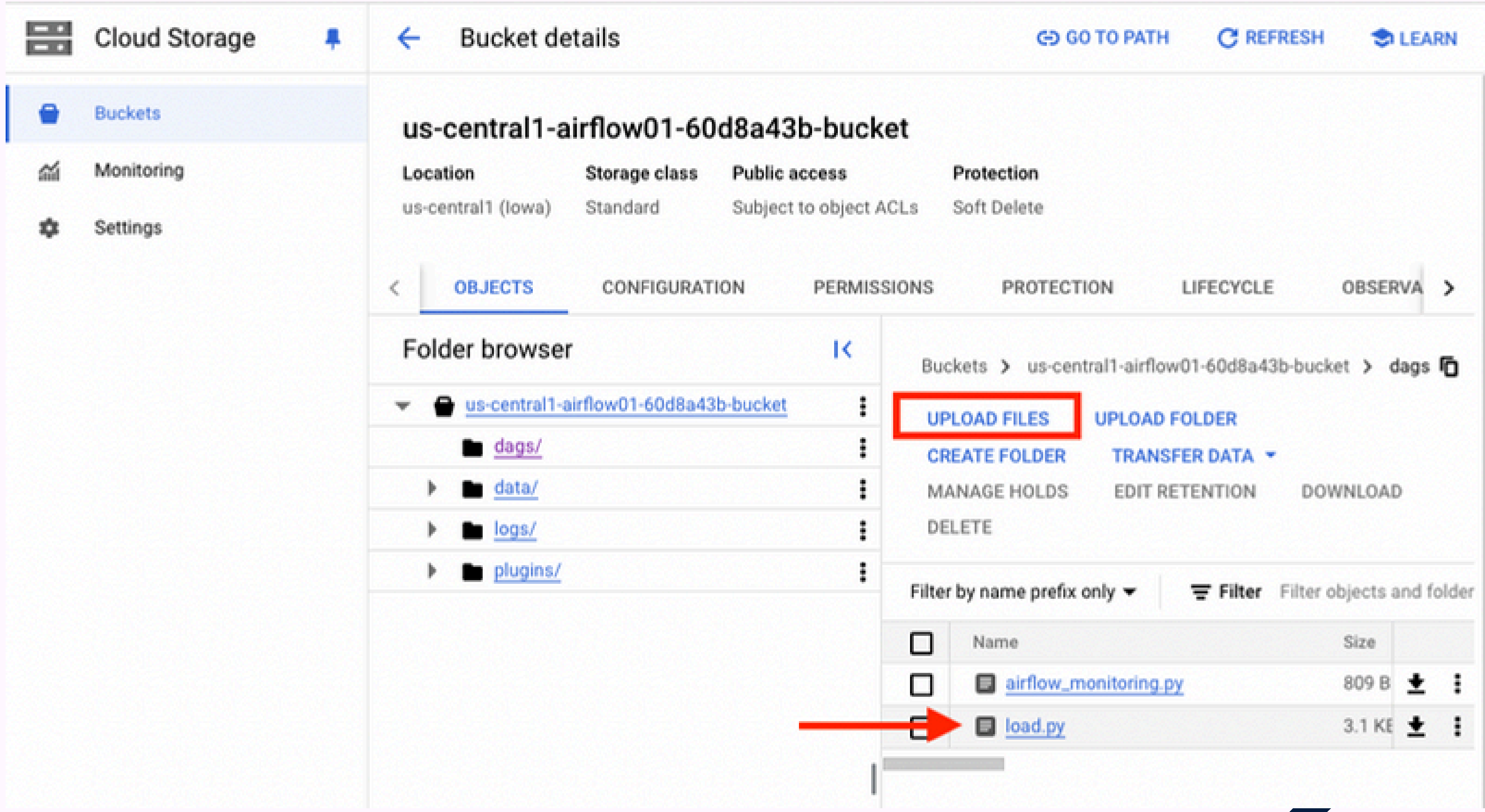
Upload DAG file

ONE



	State	Name ↑	Location	Composer version	Airflow version	Creation time	Update time	Airflow webserver	DAG list	Logs	Gs folder	Labels
□	✓	airflow01	us-central1	3	2.7.3-build.7	7/14/24, 12:20 AM	7/14/24, 2:05 AM	Airflow	DAGs	Logs	DAGs	None

TWO

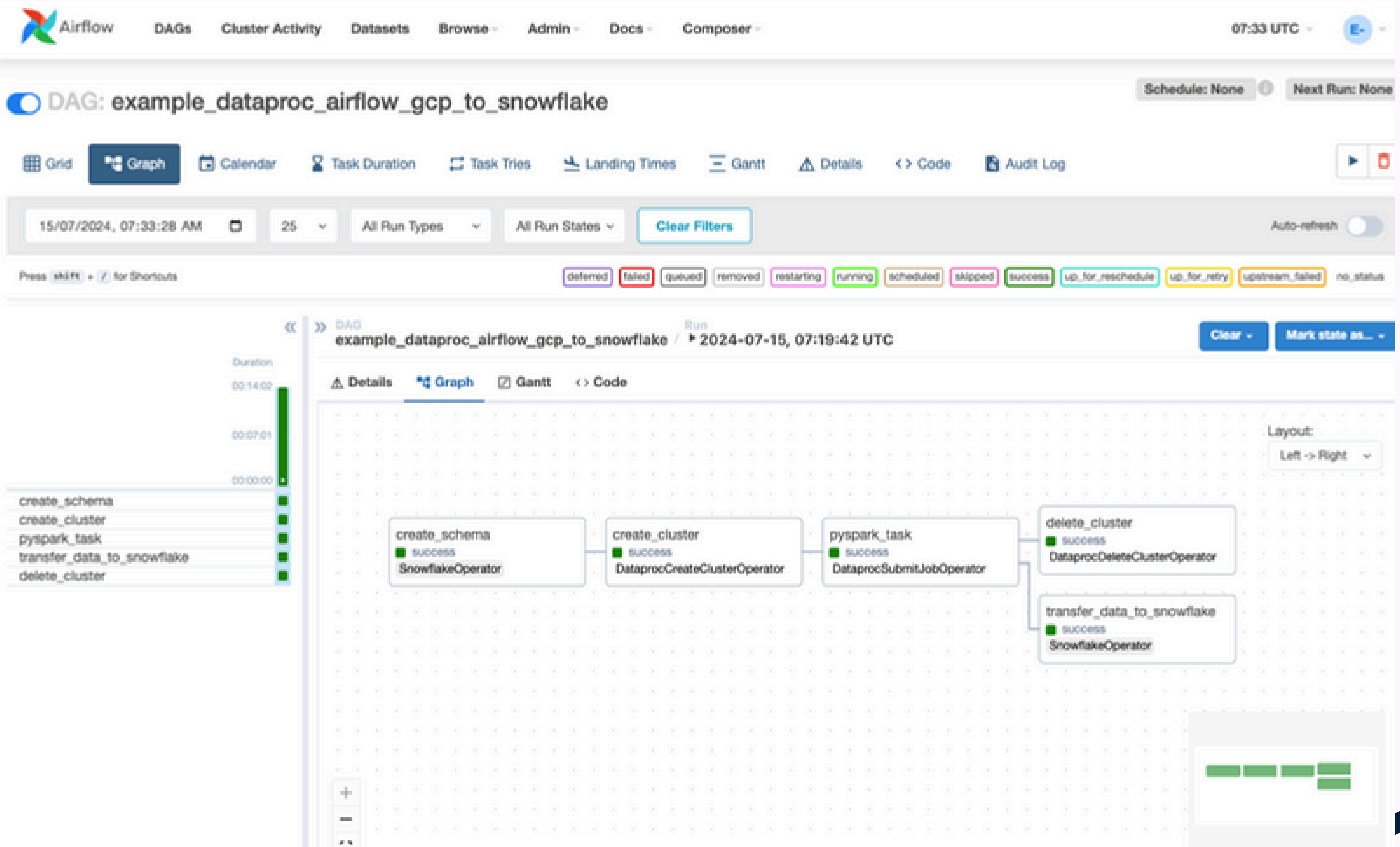


us-central1-airflow01-60d8a43b-bucket

Location	Storage class	Public access	Protection
us-central1 (Iowa)	Standard	Subject to object ACLs	Soft Delete

	Name	Size
□	airflow_monitoring.py	809 B
□	load.py	3.1 KE

Output of Airflow



DAG: example_dataproc_airflow_gcp_to_snowflake

Schedule: None

Next Run: None

15/07/2024, 07:33:28 AM | 25 | All Run Types | All Run States

Press shift + / for Shortcuts

DAG example_dataproc_airflow_gcp_to_snowflake / Run 2024-07-15, 07:19:42 UTC

Layout: Left -> Right

create_schema 🟩
create_cluster 🟩
pyspark_task 🟩
transfer_data_to_snowflake 🟩
delete_cluster 🟩

create_schema
🟩 success
SnowflakeOperator

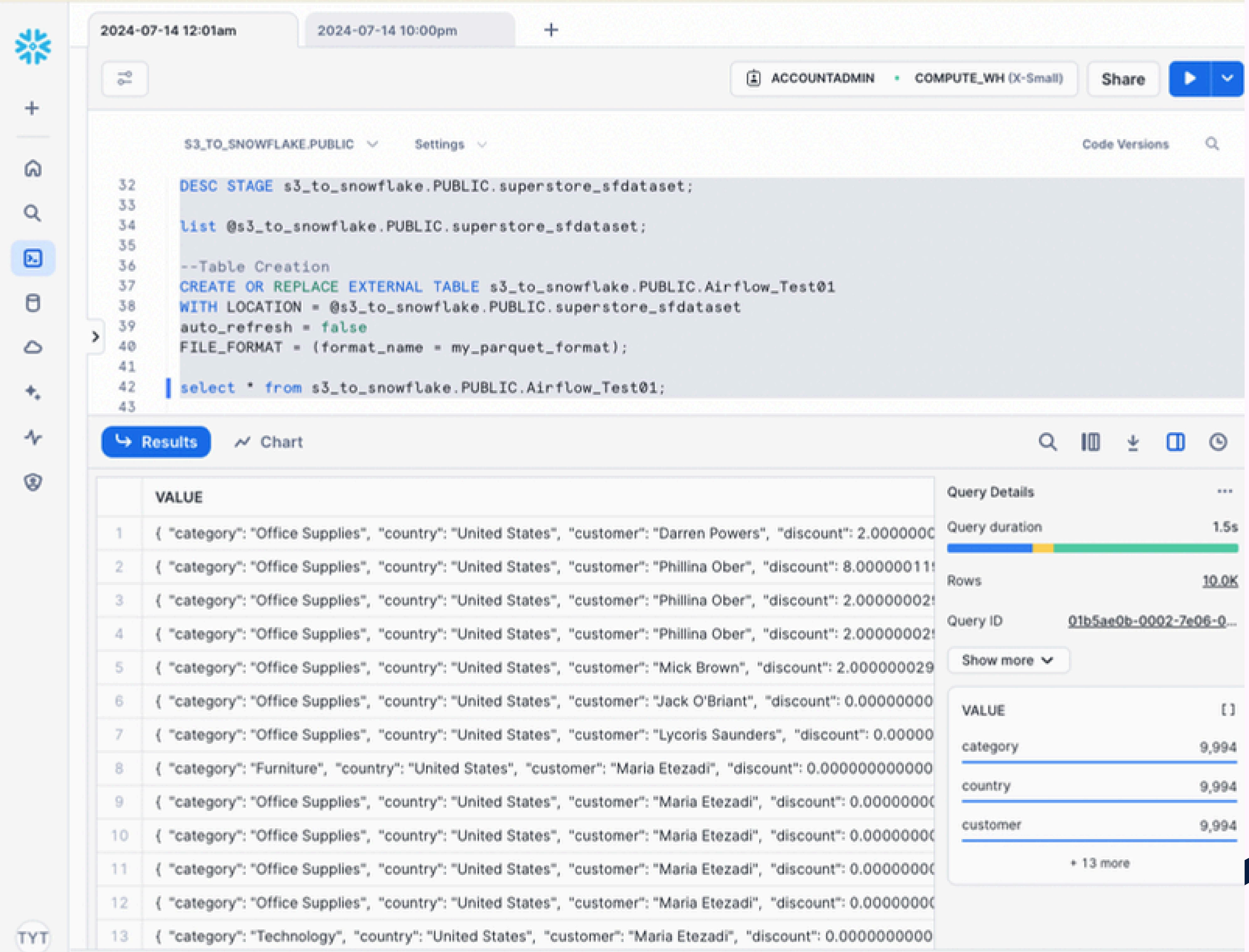
create_cluster
🟩 success
DataprocCreateClusterOperator

pyspark_task
🟩 success
DataprocSubmitJobOperator

delete_cluster
🟩 success
DataprocDeleteClusterOperator

transfer_data_to_snowflake
🟩 success
SnowflakeOperator

Output of the Snowflake



```
DESC STAGE s3_to_snowflake.PUBLIC.superstore_sfdataset;

list @s3_to_snowflake.PUBLIC.superstore_sfdataset;

--Table Creation
CREATE OR REPLACE EXTERNAL TABLE s3_to_snowflake.PUBLIC.Airflow_Test01
WITH LOCATION = @s3_to_snowflake.PUBLIC.superstore_sfdataset
auto_refresh = false
FILE_FORMAT = (format_name = my_parquet_format);

select * from s3_to_snowflake.PUBLIC.Airflow_Test01;
```

	VALUE
1	{ "category": "Office Supplies", "country": "United States", "customer": "Darren Powers", "discount": 2.000000C
2	{ "category": "Office Supplies", "country": "United States", "customer": "Phillina Ober", "discount": 8.00000011!
3	{ "category": "Office Supplies", "country": "United States", "customer": "Phillina Ober", "discount": 2.00000002!
4	{ "category": "Office Supplies", "country": "United States", "customer": "Phillina Ober", "discount": 2.00000002!
5	{ "category": "Office Supplies", "country": "United States", "customer": "Mick Brown", "discount": 2.000000029
6	{ "category": "Office Supplies", "country": "United States", "customer": "Jack O'Briant", "discount": 0.00000000
7	{ "category": "Office Supplies", "country": "United States", "customer": "Lycoris Saunders", "discount": 0.00000
8	{ "category": "Furniture", "country": "United States", "customer": "Maria Etezadi", "discount": 0.000000000000
9	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.0000000C
10	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.0000000C
11	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.0000000C
12	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.0000000C
13	{ "category": "Technology", "country": "United States", "customer": "Maria Etezadi", "discount": 0.0000000000

Query Details

Query duration	1.5s
Rows	10.0K
Query ID	01b5ae0b-0002-7e06-0...

Show more

VALUE	
category	9,994
country	9,994
customer	9,994

+ 13 more

Thank You