



雪餃 Snowflakes

Then Tsze Yen

Step 1:

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

The screenshot shows the 'Bucket details' page for a bucket named 'gcp_pipelines'. The bucket is located in 'us (multiple regions in United States)', has a 'Standard' storage class, 'Not public' access, and 'Soft Delete' protection. The 'OBJECTS' tab is selected, showing a folder browser on the left with 'gcp_pipelines' expanded to show subfolders: 'code/', 'input_files/', and 'output_files/'. On the right, there are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'TRANSFER DATA', 'MANAGE HOLDS', 'EDIT RETENTION', 'DOWNLOAD', and 'DELETE'. Below these buttons is a table of objects, which currently shows only folders.

<input type="checkbox"/>	Name	Size	Type	Created	Storage
<input type="checkbox"/>	code/	—	Folder	—	—
<input type="checkbox"/>	input_files/	—	Folder	—	—
<input type="checkbox"/>	output_files/	—	Folder	—	—

Step 2 :

Create a Cloud Storage Integration in Snowflake

The screenshot displays the Snowflake web interface. On the left, the navigation pane shows the 'Databases' tab selected, with a search bar and a list of objects including 'SNOWFLAKE' and 'SNOWFLAKE_SAMPLE_DATA'. The main panel shows the SQL editor with the following code:

```
1 CREATE OR REPLACE STORAGE INTEGRATION gcp_integration
2   TYPE = EXTERNAL_STAGE
3   STORAGE_PROVIDER = GCS
4   ENABLED = TRUE
5   STORAGE_ALLOWED_LOCATIONS = ('gcs://gcp_pipelines/');
6
7 DESC STORAGE INTEGRATION gcp_integration;
```

Below the SQL editor, the 'Results' tab is active, showing a single row of results:


status
1 Integration GCP_INTEGRATION successfully created.


On the right side of the interface, the 'Query Details' panel provides additional information:


- Query duration: 159ms
- Rows: 1
- Query ID: 01b5a498-0002-7b75-0...
- Show more
- status: 100% filled


Step 3:


Create Role


 IAM & Admin


 PAM NEW


 Principal Access Boundary


 Identity & Organization


 Policy Troubleshooter


 Policy Analyzer NEW


 Organization Policies


 Service Accounts


 Workload Identity Federat...


 Workforce Identity Federa...


 Labels


 Tags


 Settings


 Privacy & Security

 Identity-Aware Proxy

 Roles

 Audit Logs

 Manage Resources

 Edit 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

+ ADD PERMISSIONS

5 assigned permissions

Filter

Enter property name or value

☒

Permission

↑

☒

storage.buckets.get

Supported

☒

storage.objects.create

Supported

☒

storage.objects.delete

Supported

☒

storage.objects.get

Supported

☒

storage.objects.list

Supported

2024-07-14 12:01am 2024-07-14 10:00pm +

ACCOUNTADMIN * COMPUTE_WH (X-Small) Share

S3_TO_SNOWFLAKE.PUBLIC Settings Code Versions

```
11 -- Create storage integration (run this once, outside of the main script)
12 CREATE OR REPLACE STORAGE INTEGRATION S3_INTEGRATION
13   TYPE = EXTERNAL_STAGE
14   STORAGE_PROVIDER = GCS
15   ENABLED = TRUE
16   STORAGE_ALLOWED_LOCATIONS = ('gcs://gcp_pipelines/output_files/');
17
18 -- Describe the integration to get the service account email
19 DESC INTEGRATION S3_INTEGRATION;
20
21 -- GRANT USAGE ON INTEGRATION S3_INTEGRATION TO ROLE accountadmin;
22
```

Results Chart

	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.iam.gserviceaccount.com	
6	COMMENT	String		

Google Cloud Cloud Storage

Buckets Monitoring Settings

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

The screenshot shows the Google Cloud Composer console. At the top, there's a table of environments with one entry, 'airflow01', in a 'Running' state. Below this, the 'Environment details' page for 'airflow01' is shown. A message states: 'While this environment is being updated, it cannot be edited or deleted.' Below that, a red error box displays a traceback for a 'Broken DAG (dags/load(snowflakes).py)'. The error is a 'ModuleNotFoundError: No module named 'airflow.providers.snowflake''. A red arrow points from this error to the 'PYPI PACKAGES' tab in the bottom navigation bar, which is also highlighted with a red box. The 'PYPI PACKAGES' tab shows 'No PyPI packages added'.

The screenshot shows a Stack Overflow answer by Basem Gaber. The answer is marked as the accepted solution with a green checkmark. It describes the problem: '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.' The solution provided is: '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.' The answer was posted on Jun 9, 2022, and edited on Jun 11, 2022. It has 43 votes and 7 comments.

Airflow

DAGs

Cluster Activity

Datasets

Browse

Admin

Docs

Composer

TWO

Canada

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

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

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

Save

Test

Airflow

DAGs

Cluster Activity

Datasets

Browse

Admin

Docs

Composer

ONE

Canada

List Connection

Variables

Configurations

Connections

Plugins

Pools

XComs

Search

+

Actions

-

Record Count: 6

	Conn Id	Conn Type	Description	Host	Port	Is Encrypted	Is Extra Encrypted
<input type="checkbox"/>	airflow_db	postgres		airflow-sqiproxy-service.composer-system.svc.cluster.local	3306	True	False
<input type="checkbox"/>	bigquery_default	google_cloud_platform				False	True
<input type="checkbox"/>	google_cloud_datastore_default	google_cloud_platform				False	True
<input type="checkbox"/>	google_cloud_default	google_cloud_platform				False	True
<input type="checkbox"/>	google_cloud_storage_default	google_cloud_platform				False	True
<input type="checkbox"/>	snowflake_conn	snowflake				True	True

Step 6:

Add airflow-snowflake connection

Step 6:

Upload DAG file

ONE

Google Cloud | airflow-workings

Composer | Environments

State	Name	Location	Composer version	Airflow version	Creation time	Update time	Airflow webserver	DAG list	Logs	Gs folder	Labels
<input type="checkbox"/>	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

Cloud Storage | Bucket details

us-central1-airflow01-60d8a43b-bucket

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

OBJECTS | CONFIGURATION | PERMISSIONS | PROTECTION | LIFECYCLE | OBSERVE

Folder browser

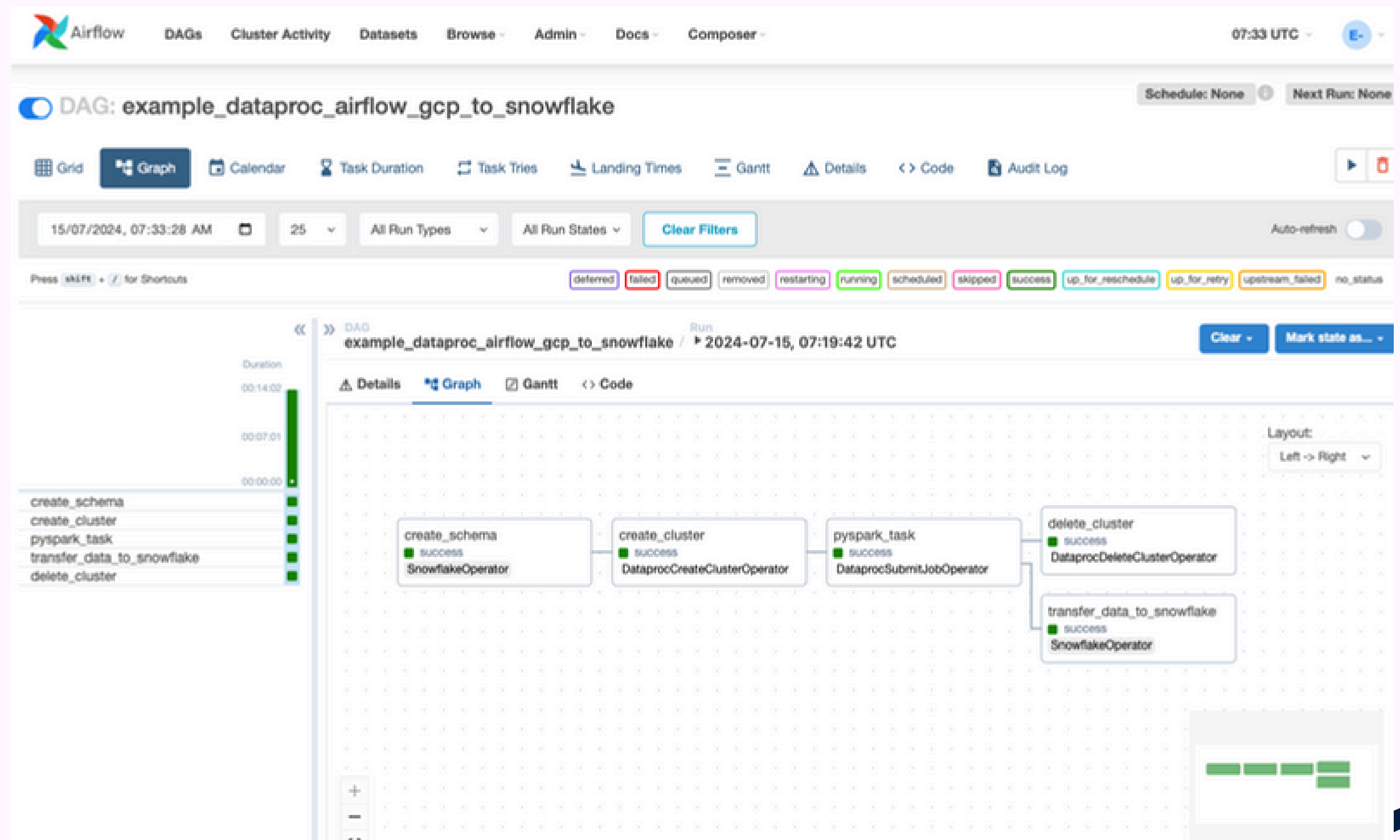
- us-central1-airflow01-60d8a43b-bucket
 - dags/
 - data/
 - logs/
 - plugins/

Actions: **UPLOAD FILES**, UPLOAD FOLDER, CREATE FOLDER, TRANSFER DATA, MANAGE HOLDS, EDIT RETENTION, DOWNLOAD, DELETE

Filter by name prefix only

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

Output of Airflow



Output of the Snowflake

The screenshot displays the Snowflake SQL Editor interface. At the top, there are tabs for the current session (2024-07-14 12:01am) and a previous session (2024-07-14 10:00pm). The user is logged in as ACCOUNTADMIN on the COMPUTE_WH (X-Small) warehouse. The SQL editor shows a query with the following lines:

```
32 DESC STAGE s3_to_snowflake.PUBLIC.superstore_sfdataset;  
33  
34 list @s3_to_snowflake.PUBLIC.superstore_sfdataset;  
35  
36 --Table Creation  
37 CREATE OR REPLACE EXTERNAL TABLE s3_to_snowflake.PUBLIC.Airflow_Test01  
38 WITH LOCATION = @s3_to_snowflake.PUBLIC.superstore_sfdataset  
39 auto_refresh = false  
40 FILE_FORMAT = (format_name = my_parquet_format);  
41  
42 select * from s3_to_snowflake.PUBLIC.Airflow_Test01;  
43
```

Below the editor, the 'Results' tab is active, showing a table with 13 rows. The first row is the header 'VALUE'. The subsequent rows contain JSON objects representing data from the 'superstore_sfdataset' stage. The 'Query Details' panel on the right shows a query duration of 1.5s and 10,000 rows returned. The 'Query ID' is 01b5ae0b-0002-7e06-0...

	VALUE
1	{ "category": "Office Supplies", "country": "United States", "customer": "Darren Powers", "discount": 2.00000000
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": 2.00000002
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": 0.000000
8	{ "category": "Furniture", "country": "United States", "customer": "Maria Etezadi", "discount": 0.000000000000
9	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.00000000
10	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.00000000
11	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.00000000
12	{ "category": "Office Supplies", "country": "United States", "customer": "Maria Etezadi", "discount": 0.00000000
13	{ "category": "Technology", "country": "United States", "customer": "Maria Etezadi", "discount": 0.0000000000

Query Details

- Query duration: 1.5s
- Rows: 10,000
- Query ID: 01b5ae0b-0002-7e06-0...

VALUE []

category	count
category	9,994
country	9,994
customer	9,994

+ 13 more



*Thank
You*