

# Cloud Only Data Pipeline Implementation

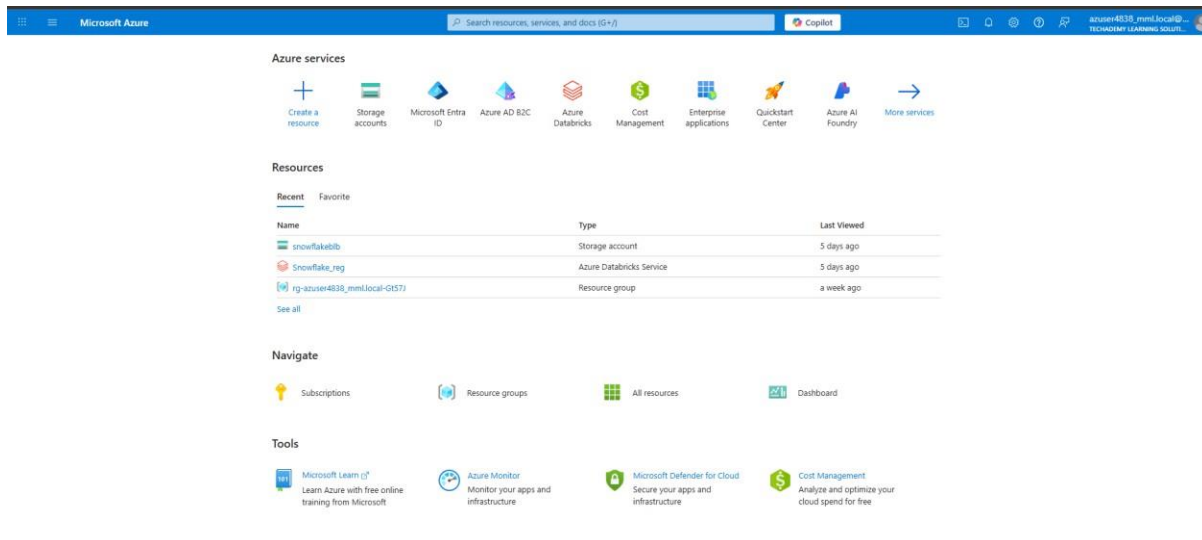
1. **Receives CSV files** from the sales team (monthly sales data).
2. **Stores them in Azure Blob Storage.**
3. **Processes and ingests them into Snowflake** using **Snowpark** (Snowflake's Python API).
4. **Transforms the data** into structured tables and views.
5. **Visualizes the data** in **Power BI** for business users.

Sales Team → Azure Blob Storage → Azure Databricks (Snowpark) → Snowflake → Power BI

## Phase 1: Setting Up Azure Blob Storage

### Step 1: Create a Storage Account

1. Sign in to the Azure Portal.



2. Click "**Create a resource**" and search for "**Storage account**".

**Create a storage account**

**Basics** | Advanced | Networking | Data protection | Encryption | Tags | Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

**Project details**  
Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription \*

Resource group \*  [Create new](#)

**Instance details**

Storage account name \*

Region \*  [Deploy to an Azure Extended Zone](#)

Preferred storage type    
 ⓘ This helps us provide relevant guidance. It doesn't restrict your storage to this resource type. [Learn more](#)

Performance ⓘ   
☒ **Standard**: Recommended for most scenarios (general-purpose v2 account)   
☐ **Premium**: Recommended for scenarios that require low latency.

Redundancy ⓘ   
   
☒ Make read access to data available in the event of regional unavailability.

[Previous](#) [Next](#) [Review + create](#) [Give feedback](#)

3. Choose your subscription and either select an existing resource group or create a new one (e.g., ItTechGenie-RG).
4. Set the storage account name (e.g., itgretailstorage) and region (e.g., East US).
5. Choose **Standard** performance and **Locally-redundant storage (LRS)** for redundancy.
6. Click **Review + Create**, then **Create**.

**ittechgeniestorage**

Storage account

Overview | Activity log | Tags | Diagnose and solve problems | Access Control (IAM) | Data migration | Events | Storage browser | Storage Mover | Partner solutions | Resource visualizer | Data storage | Security + networking | Networking | Front Door and CDN | Access keys | Shared access signature | Encryption | Microsoft Defender for Cloud | Data management | Storage Actions | Redundancy | Data protection

**Essentials**

Resource group (mou5) : rg-azuser4838\_mml-local-Q157

Location : centralindia

Primary/Secondary Location : Primary: Central India, Secondary: South India

Subscription (mou5) : MML Learners

Subscription ID : 2a3c6418-9789-4d96-a24b-2c2d7633d375

Disk state : Primary: Available, Secondary: Available

Tags (edit) : Add tags

**Properties** | Monitoring | Capabilities (7) | Recommendations (0) | Tutorials | Tools + SDKs

**Blob service**

Hierarchical namespace	Disabled
Default access tier	Hot
Blob anonymous access	Disabled
Blob soft delete	Disabled
Container soft delete	Disabled
Versioning	Disabled
Change feed	Disabled
NFS v3	Disabled
Allow cross-tenant replication	Disabled
Storage tasks assignments	None

**File service**

Large file share	Enabled
Identity-based access	Not configured
Default share-level permissions	Disabled
Soft delete	Disabled

**Security**

Require secure transfer for REST API operations	Enabled
Storage account key access	Enabled
Minimum TLS version	Version 1.2
Infrastructure encryption	Disabled

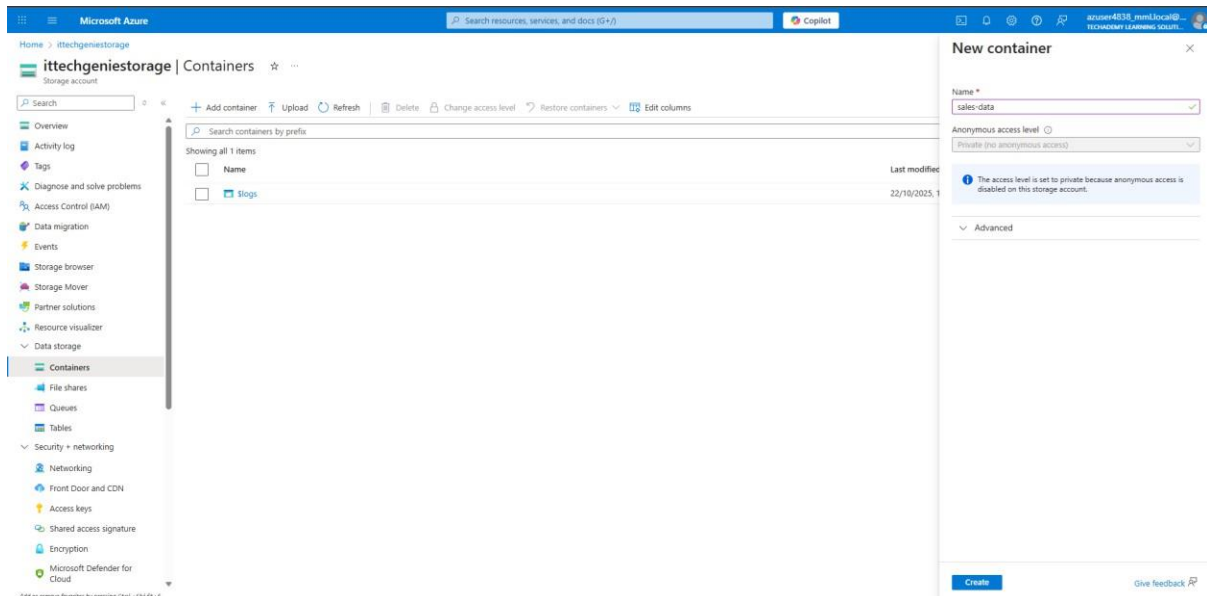
**Networking**

Public network access	Enabled
Public network access scope	Enable from all networks
Private endpoint connections	0
Network routing	Microsoft network routing
Endpoint type	Standard

## Step 2: Create a Container

1. Inside your storage account, go to **"Containers"**.

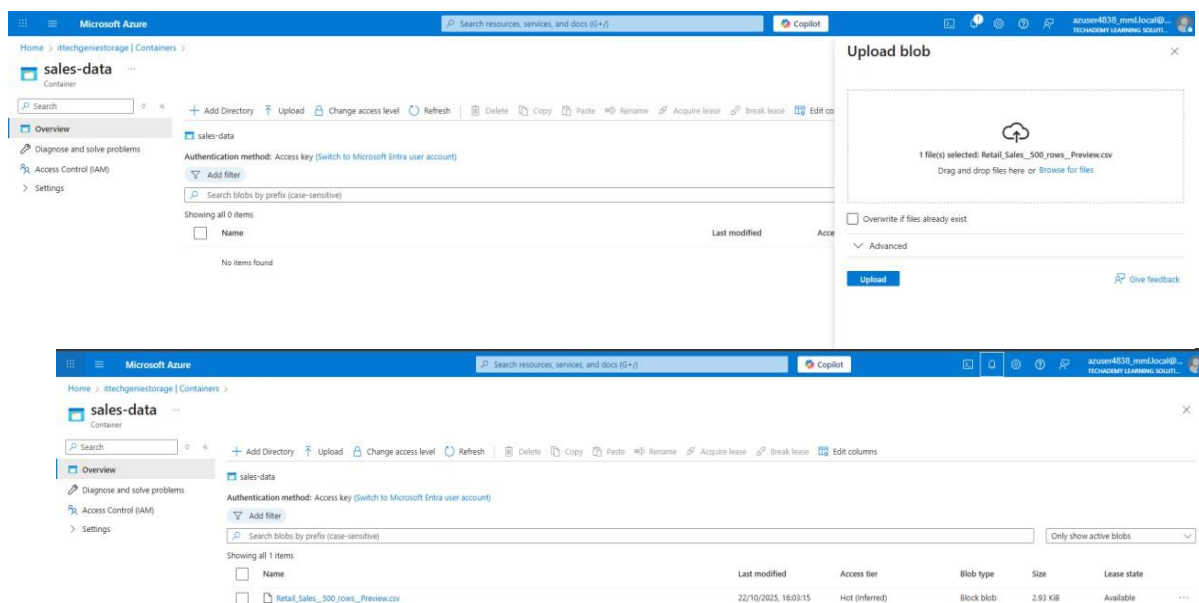
2. Click **" + Container "** and name it something like `monthly-sales`.
3. Set the access level to **Private**.



## Step 3: Upload the CSV File

### Option A: Azure Portal

- Navigate to the `monthly-sales` container.
- Click **Upload**, select your .csv file (e.g., `sales_october.csv`), and upload it.



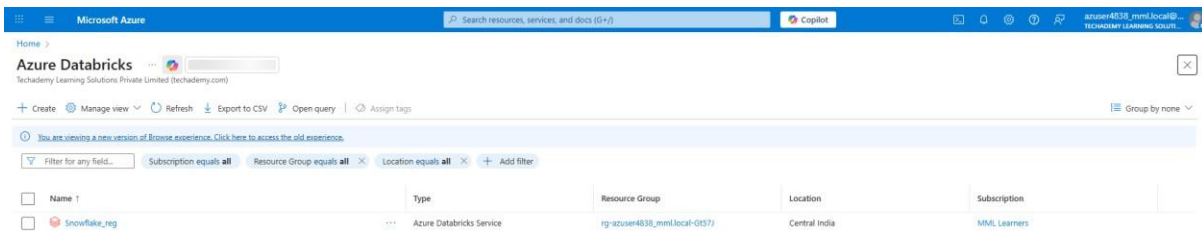
## Option B: Azure Cloud Shell

```
az storage blob upload \  
--account-name itgretailstorage \  
--container-name monthly-sales \  
--name sales_october.csv \  
--file sales_october.csv \  
--auth-mode login
```

## Phase 2: Provisioning Azure Databricks

### Step 1: Create a Databricks Workspace

1. In Azure Portal, click **Create a resource** and search for **Azure Databricks**.

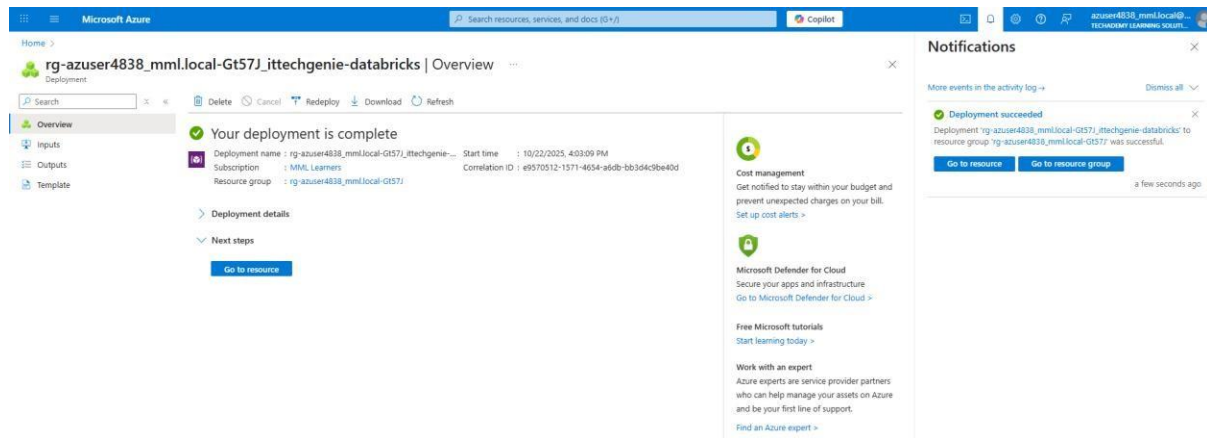


2. Set the workspace name (e.g., itgretail-databricks) and region (same as storage).
3. Choose the **Premium** pricing tier.
4. Click **Review + Create**, then **Create**.

A screenshot of the 'Create an Azure Databricks workspace' form in the Azure Portal. The form is divided into sections: 'Project Details' and 'Instance Details'. In the 'Project Details' section, the 'Subscription' is set to 'MMML Learners' and the 'Resource group' is 'rg-azurer4838\_mmmllocal-G457'. In the 'Instance Details' section, the 'Workspace name' is 'ittechgenie-databricks', the 'Region' is 'Central India', and the 'Pricing Tier' is 'Premium (+ Role-based access controls)'. The 'Managed Resource Group name' field is empty. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next: Networking >'.

Project Details	
Subscription *	MMML Learners
Resource group *	rg-azurer4838_mmmllocal-G457

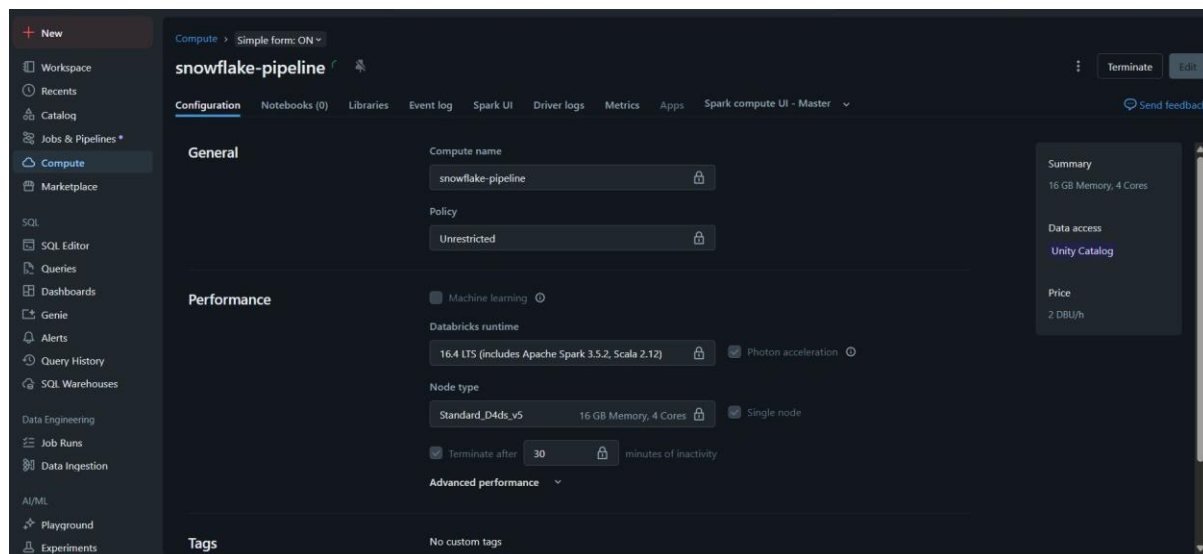
Instance Details	
Workspace name *	ittechgenie-databricks
Region *	Central India
Pricing Tier *	Premium (+ Role-based access controls)
Managed Resource Group name	Enter name for managed resource group



## Step 2: Launch and Configure a Cluster

1. Open the Databricks workspace.
2. Go to **Compute** and click **Create Cluster**.
3. **Configure:**

- Name: snowflake-pipeline
- Mode: Single Node
- Runtime: 12.2 LTS
- Node Type: Standard\_DS3\_v2
- Auto-termination: 30 minutes



## Phase 3: Snowflake Setup

### Step 1: Create Snowflake Objects

Run the following SQL in Snowflake

```
CREATE WAREHOUSE ITG_WAREHOUSE WITH WAREHOUSE_SIZE = XSMALL  
AUTO_SUSPEND = 300 AUTO_RESUME = TRUE;
```

```
CREATE DATABASE ITG_SALES_DB;  
CREATE SCHEMA ITG_SALES_DB.RAW;  
CREATE SCHEMA ITG_SALES_DB.MODELED;  
CREATE SCHEMA ITG_SALES_DB.REPORTING;
```

```
CREATE ROLE DATA_ENGINEER;  
GRANT USAGE ON WAREHOUSE ITG_WAREHOUSE TO ROLE DATA_ENGINEER;  
GRANT ALL ON DATABASE ITG_SALES_DB TO ROLE DATA_ENGINEER;
```

## **Step 2: Connect Snowflake to Azure**

```
CREATE STORAGE INTEGRATION azure_sales_integration  
  TYPE = EXTERNAL_STAGE  
  STORAGE_PROVIDER = AZURE  
  ENABLED = TRUE  
  AZURE_TENANT_ID = '<your-tenant-id>'  
  STORAGE_ALLOWED_LOCATIONS =  
  ('azure://itgretailstorage.blob.core.windows.net/monthly-sales/');
```

**Then run:**

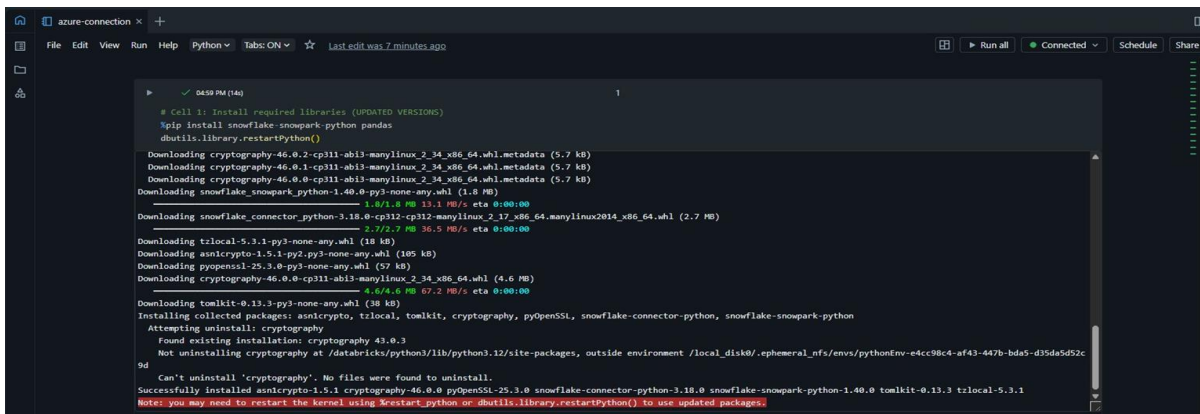
```
DESC STORAGE INTEGRATION azure_sales_integration;
```

Use the AZURE\_CONSENT\_URL to authorize Snowflake to access your Azure storage.

## **Phase 4: Data Ingestion with Snowpark in Databricks**

### **Step 1: Install Required Libraries**

```
%pip install snowflake-snowpark-python pandas  
dbutils.library.restartPython()
```



```
# Cell 1: Install required libraries (UPDATED VERSIONS)
!pip install snowflake-snowpark-python pandas
dbutils.library.restartPython()

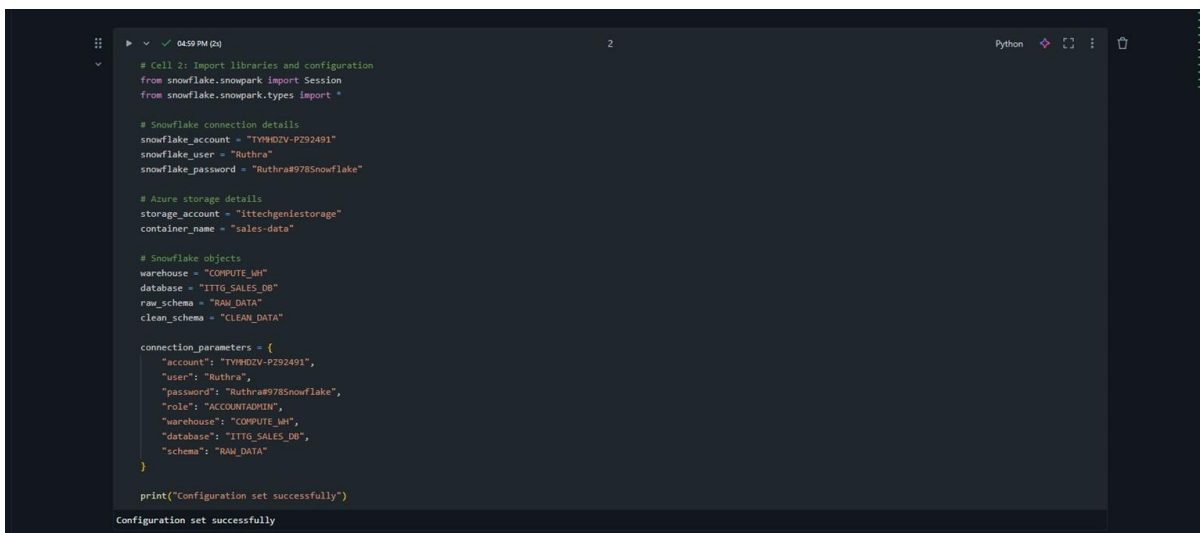
Downloading cryptography-46.0.2-cp311-ab13-manylinux_2_34_x86_64.whl.metadata (5.7 kB)
Downloading cryptography-46.0.1-cp311-ab13-manylinux_2_34_x86_64.whl.metadata (5.7 kB)
Downloading cryptography-46.0.0-cp311-ab13-manylinux_2_34_x86_64.whl.metadata (5.7 kB)
Downloading snowflake_snowpark_python-1.40.0-py3-none-any.whl (1.8 MB)
----- 1.8/1.8 MB 13.1 MB/s eta 0:00:00
Downloading snowflake_connector_python-3.18.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.7 MB)
----- 2.7/2.7 MB 36.5 MB/s eta 0:00:00
Downloading tlocal-5.3.1-py3-none-any.whl (18 kB)
Downloading asnycrypto-1.5.1-py2.py3-none-any.whl (185 kB)
Downloading pyopenssl-25.3.0-py3-none-any.whl (57 kB)
Downloading cryptography-46.0.0-cp311-ab13-manylinux_2_34_x86_64.whl (4.6 MB)
----- 4.6/4.6 MB 67.2 MB/s eta 0:00:00
Downloading tomkit-0.13.3-py3-none-any.whl (38 kB)
Installing collected packages: asnycrypto, tlocal, tomkit, cryptography, pyOpenSSL, snowflake-connector-python, snowflake-snowpark-python
Attempting uninstall: cryptography
Found existing installation: cryptography 43.0.3
NOT uninstalling cryptography at /databricks/python3/lib/python3.12/site-packages, outside environment /local_disk0/.ephemeral_nfs/venv/pythonEnv-e4cc98c4-af43-447b-bda5-d35da5d52c9d
Can't uninstall 'cryptography'. No files were found to uninstall.
Successfully installed asnycrypto-1.5.1 cryptography-46.0.0 pyOpenSSL-25.3.0 snowflake-connector-python-3.18.0 snowflake-snowpark-python-1.40.0 tomkit-0.13.3 tlocal-5.3.1
Note: you may need to restart the kernel using %restart_python or dbutils.library.restartPython() to use updated packages.
```

## Step 2: Configure Snowflake Connection

from snowflake.snowpark import Session

```
connection_parameters = {
    "account": "<your_account>",
    "user": "<your_user>",
    "password": "<your_password>",
    "role": "ACCOUNTADMIN",
    "warehouse": "ITG_WAREHOUSE",
    "database": "ITG_SALES_DB",
    "schema": "RAW"
}
```

session = Session.builder.configs(connection\_parameters).create()



```
# Cell 2: Import libraries and configuration
from snowflake.snowpark import Session
from snowflake.snowpark.types import *

# Snowflake connection details
snowflake_account = "TYMHQZV-P292491"
snowflake_user = "Ruthra"
snowflake_password = "Ruthra@978Snowflake"

# Azure storage details
storage_account = "littechgeniostorage"
container_name = "sales-data"

# Snowflake objects
warehouse = "COMPUTE_WH"
database = "ITG_SALES_DB"
raw_schema = "RAW_DATA"
clean_schema = "CLEAN_DATA"

connection_parameters = {
    "account": "TYMHQZV-P292491",
    "user": "Ruthra",
    "password": "Ruthra@978Snowflake",
    "role": "ACCOUNTADMIN",
    "warehouse": "COMPUTE_WH",
    "database": "ITG_SALES_DB",
    "schema": "RAW_DATA"
}

print("Configuration set successfully")

Configuration set successfully
```

## Step 3: Create Stage and File Format

```

session.sql("""
CREATE OR REPLACE FILE FORMAT csv_format
  TYPE = 'CSV'
  FIELD_DELIMITER = ','
  SKIP_HEADER = 1
  NULL_IF = ('NULL', 'null')
  EMPTY_FIELD_AS_NULL = TRUE;
""").collect()

```

```

session.sql("""
CREATE OR REPLACE STAGE azure_stage
  URL = 'azure://itgretailstorage.blob.core.windows.net/monthly-sales/'
  CREDENTIALS = (AZURE_SAS_TOKEN = '<your-sas-token>')
  FILE_FORMAT = csv_format;
""").collect()

```

```

session.sql("""
CREATE OR REPLACE FILE FORMAT csv_sales_format
  TYPE = 'CSV'
  FIELD_DELIMITER = ','
  SKIP_HEADER = 1
  NULL_IF = ('NULL', 'null')
  EMPTY_FIELD_AS_NULL = TRUE;
""").collect()

session.sql("""
CREATE OR REPLACE STAGE azure_sales_stage
  URL = 'azure://ittechgeniestorage.blob.core.windows.net/sales-data/'
  CREDENTIALS = (
    AZURE_SAS_TOKEN = '/?sp=racwld&st=2025-10-22T10:47:09Z&se=2025-10-23T19:02:09Z&spr=https&sv=2024-11-04&sr=cksig=HEF760inEZPKZBybuk9F2FvtAou82f3N2BoDvfc3fNQ5flbsK3D'
  )
  FILE_FORMAT = csv_sales_format;
""").collect()

print("File format and stage created successfully")

```

File format and stage created successfully

## Step 4: Load Data into Snowflake

```

session.sql("""
COPY INTO raw_sales_data (
  OrderID, OrderDate, MonthOfSale, CustomerID, CustomerName,
  Country, Region, City, Category, Subcategory,
  Quantity, Discount, Sales, Profit, FileName
)
FROM (
  SELECT $1, $2, $3, $4, $5, $6, $7, $8, $9, $10,
    $11, $12, $13, $14, METADATA$FILENAME
  FROM @azure_stage/sales_october.csv
)
FILE_FORMAT = (FORMAT_NAME = csv_format)
ON_ERROR = 'CONTINUE';
""").collect()

```



```
result = session.sql("SELECT COUNT(*) as total_rows FROM raw_sales_data").collect()
print(f"Total rows in raw table: {result[0]['TOTAL_ROWS']}")

print("Sample raw data:")
session.sql("SELECT * FROM raw_sales_data LIMIT 5").show()
```

Total rows in raw table: 25  
Sample raw data:

"ORDERID"	"ORDERDATE"	"MONTHOFSALE"	"CUSTOMERID"	"CUSTOMERNAME"	"COUNTRY"	"REGION"	"CITY"	"CATEGORY"	"SUBCATEGORY"	"QUANTITY"	"DISCOUNT"	"SALES"	"PROMOTIONID"
ORD-SFBD6FOC	2024-10-08	2024-10	CUST1000	Ananya Sharma	India	South	Mumbai	Office Supplies	Paper	9	0.00	2700.00	78
ORD-BF0078E4	2024-08-11	2024-08	CUST1001	Aarav Iyer	India	Central	Lucknow	Technology	Networking	4	0.15	27200.00	41
ORD-86C2D3A3	2024-06-12	2024-06	CUST1002	Arjun Sharma	USA	East	Kolkata	Furniture	Tables	4	0.10	31500.00	56
ORD-FB8CD2D9	2024-12-10	2024-12	CUST1003	Ananya Das	India	North	Kolkata	Office Supplies	Appliances	9	0.00	36000.00	11
ORD-EF355968	2024-10-27	2024-10	CUST1004	Ishaan Bhat	UK	Central	Chennai	Furniture	Storage	4	0.00	24000.00	41

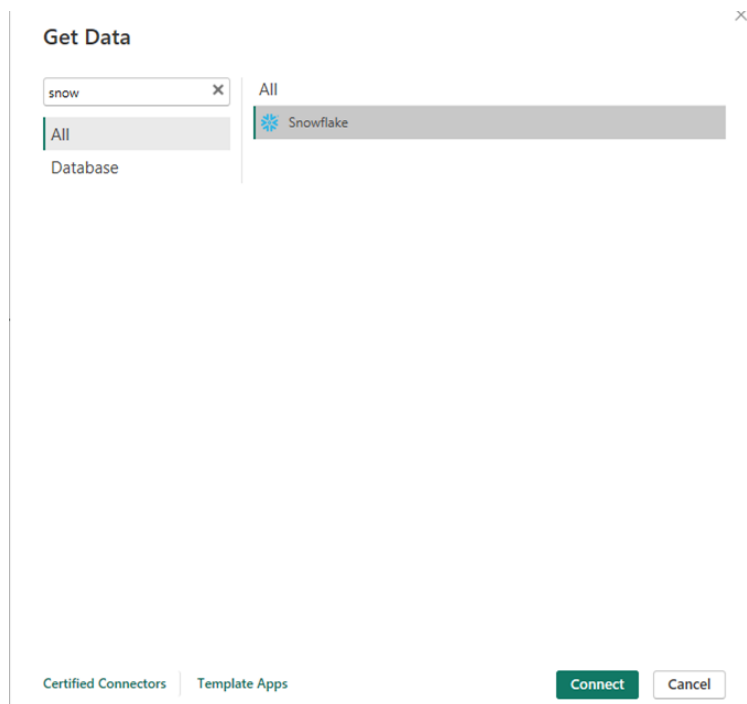
```
session.close()
print("Snowpark session closed")
print("Pipeline execution completed successfully!")
print("\nNext steps: Connect Power BI to Snowflake using:")
print("Database: ITTG_SALES_DB")
print("Schema: CLEAN_DATA")
print("View: VW_POWERBI_SALES_DASHBOARD")
```

Snowpark session closed  
Pipeline execution completed successfully!

Next steps: Connect Power BI to Snowflake using:  
Database: ITTG\_SALES\_DB  
Schema: CLEAN\_DATA  
View: VW\_POWERBI\_SALES\_DASHBOARD

## Phase 5: Power BI Reporting

1. Open Power BI Desktop.
2. Choose **Snowflake** as the data source.
3. Connect using your Snowflake credentials and select the ITG\_SALES\_DB. REPORTING schema.



#### 4. Import Data

**Navigator**

Display Options ▾

- tymhdzv-pz92491.snowflakecomputing.co...
  - ANALYTICS\_DB
  - AZURE\_SNOWPIPE\_DB
  - DEMO\_DS
  - ITTG\_SALES\_DB [5]
    - ANALYTICS
    - CLEAN\_DATA [3]
      - ☐ SALES\_SUMMARY\_MONTHLY
      - ☐ VW\_POWERBI\_SALES\_DASHBOARD
      - ☒ CLEAN\_SALES\_DATA
    - INFORMATION\_SCHEMA
    - PUBLIC
    - RAW\_DATA [3]
      - ☐ CLEAN\_SALES\_DATA
      - ☐ VW\_POWERBI\_DASHBOARD
      - ☐ RAW\_SALES\_DATA
    - MANAGE\_DB
    - MIGRATED\_DB
    - MY\_PRACTICE\_DB

**CLEAN\_SALES\_DATA**

ORDERID	ORDERDATE	MONTHOFSALE	CUSTOMERID	CUSTOMERNAME	COUNTRY	REGION	CITY	CA
ORD-5F8D6F0C	08-10-2024	2024-10	CUST1000	Ananya Sharma	India	South	Mumbai	
ORD-BF0078E4	11-08-2024	2024-08	CUST1001	Aarav Iyer	India	Central	Lucknow	
ORD-86CD58A3	12-06-2024	2024-06	CUST1002	Arjun Sharma	USA	East	Kolkata	
ORD-FB0CD2D9	18-12-2024	2024-12	CUST1003	Ananya Das	India	North	Kolkata	
ORD-EF35596B	27-10-2024	2024-10	CUST1004	Ishaan Bhat	UK	Central	Chennai	
ORD-60D1DA88	26-08-2024	2024-08	CUST1005	Neha Iyer	UAE	West	Chennai	
ORD-A5081404	15-09-2025	2025-09	CUST1006	Arjun Iyer	India	Central	Jaipur	
ORD-E1C9BE42	27-02-2024	2024-02	CUST1007	Priya Singh	India	North	Lucknow	
ORD-4FCB3B05	26-05-2025	2025-05	CUST1008	Kabir Menon	India	West	Jaipur	
ORD-921966C8	14-03-2025	2025-03	CUST1009	Arjun Chopra	UAE	West	Mumbai	
ORD-E4A002F0	12-06-2024	2024-06	CUST1010	Ananya Patel	UK	West	Ahmedabad	
ORD-0944D71F	20-07-2025	2025-07	CUST1011	Sanjay Gupta	UAE	North	Pune	
ORD-7E28FF54	21-03-2025	2025-03	CUST1012	Ananya Khan	India	North	Delhi	
ORD-55961D4C	15-09-2025	2025-09	CUST1013	Neha Mehta	India	East	Ahmedabad	
ORD-6BE57CAD	06-01-2025	2025-01	CUST1014	Aarav Reddy	UAE	East	Ahmedabad	
ORD-1D9DC086	08-05-2024	2024-05	CUST1015	Ishaan Bhat	India	West	Jaipur	
ORD-9B484AF9	12-02-2025	2025-02	CUST1016	Rohan Khan	Singapore	East	Delhi	
ORD-42167295	25-08-2025	2025-08	CUST1017	Kabir Sharma	India	Central	Mumbai	
ORD-A91119D6	10-06-2025	2025-06	CUST1018	Kabir Iyer	India	West	Ahmedabad	
ORD-951CD78B	06-04-2024	2024-04	CUST1019	Sneha Menon	India	West	Mumbai	
ORD-86A75DAS	16-02-2025	2025-02	CUST1020	Ananya Verma	USA	North	Bengaluru	
ORD-6D74C638	12-09-2024	2024-09	CUST1021	Arjun Gowda	UAE	North	Jaipur	
ORD-93240C28	30-08-2024	2024-08	CUST1022	Aarav Iyer	India	West	Mumbai	

#### 5. Load the relevant tables or views.

#### 6. Build visualizations like:

- Monthly sales trends
- Top-selling categories
- Regional performance

