Cloud Only Data Pipeline Implemetation

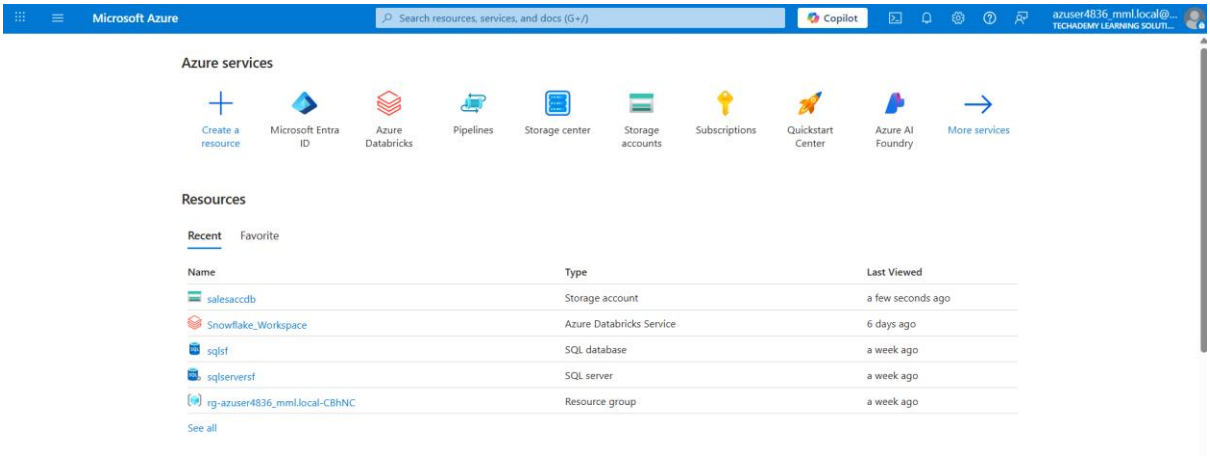
1. **Receives CSV files** from the sales team (monthly sales data).
2. **Stores them in Azure Blob Storage**.
3. **Processes and ingests them into Snowflake** using **Snowpark** (Snowflake's Python API).
4. **Transforms the data** into structured tables and views.
5. **Visualizes the data** in **Power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

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

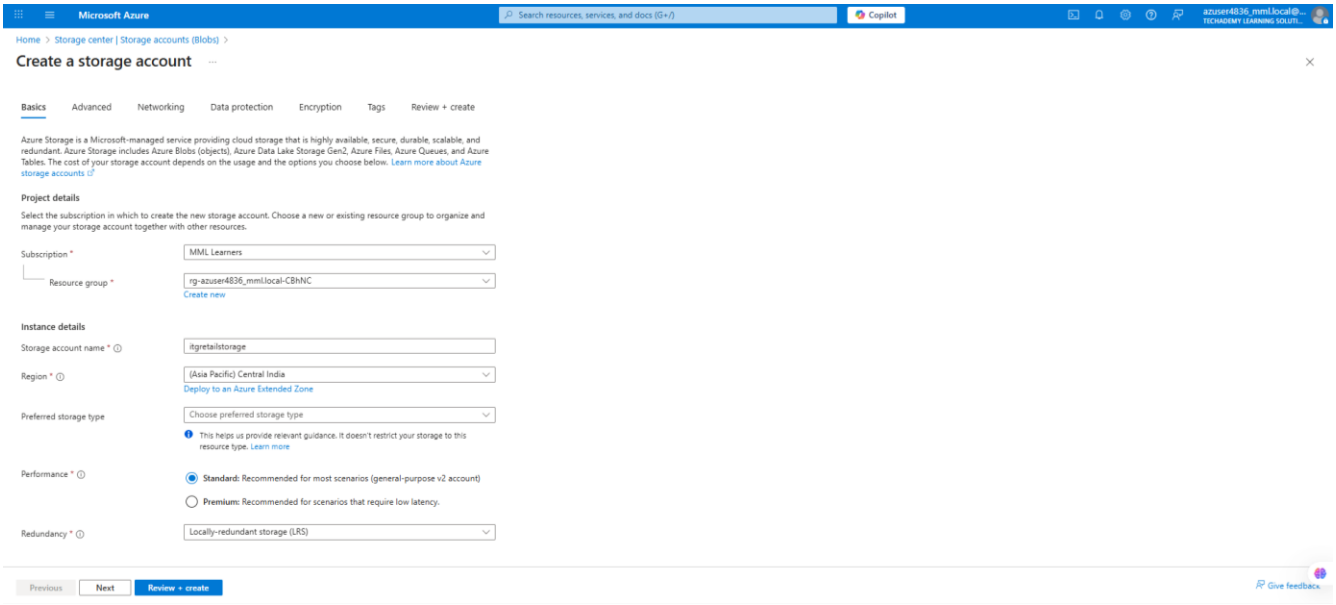
Phase 1: Setting Up Azure Blob Storage

Step 1: Create a Storage Account

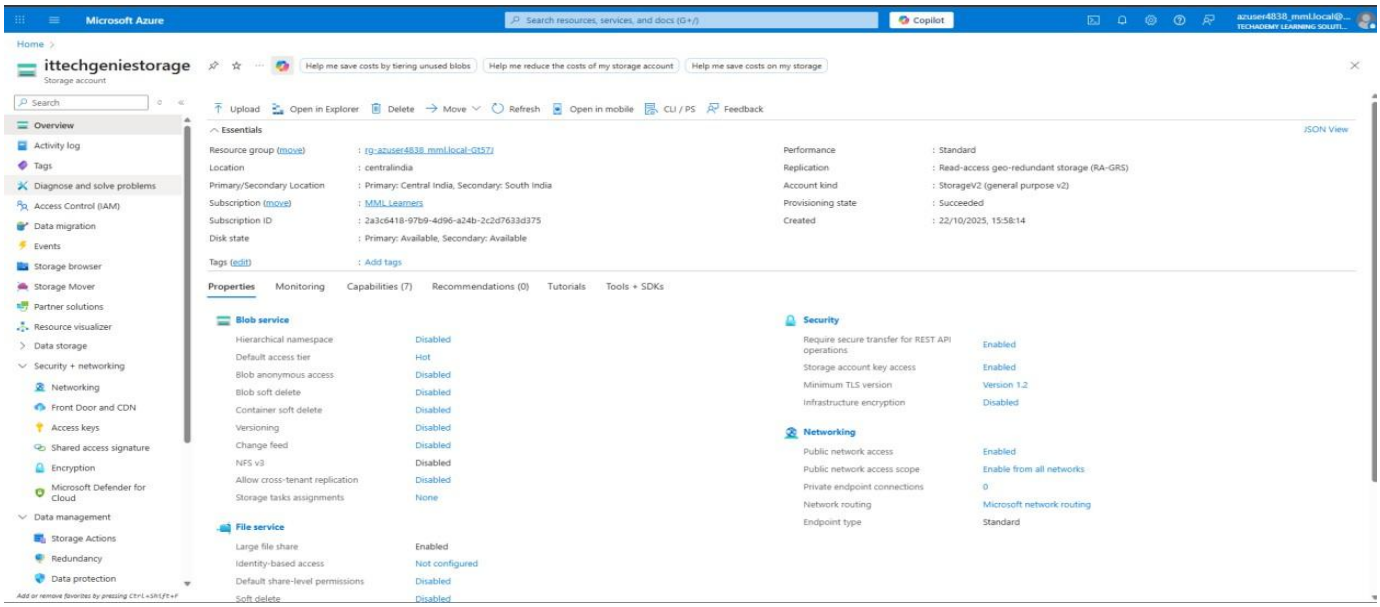
1. Sign in to the Azure Portal.

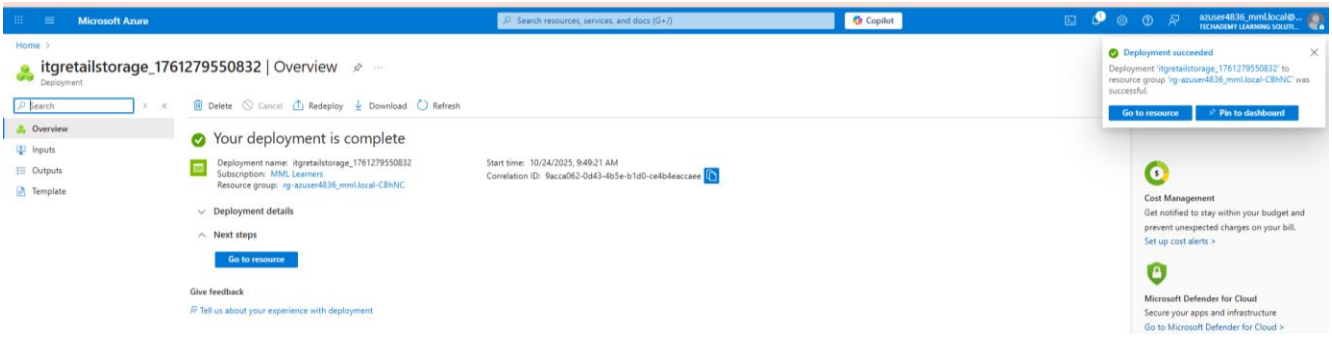


2. Click **"Create a resource"** and search for **"Storage account"**.



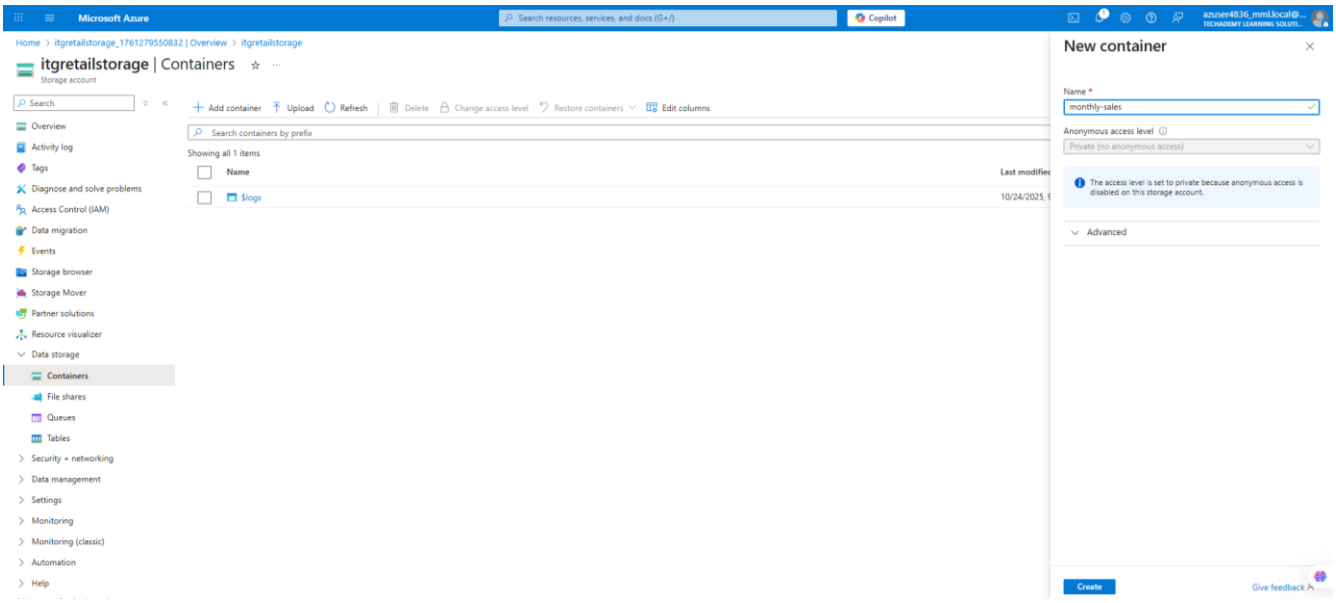
3. Choose your subscription and either select an existing resource group or create a new one
4. Set the storage account name (e.g., itgretailstorage) and region (e.g., East US).
5. Choose **Standard** performance and **Locally-redundant storage (LRS)** for redundancy.
6. Click **Review + Create**, then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





Step 2: Create a Container

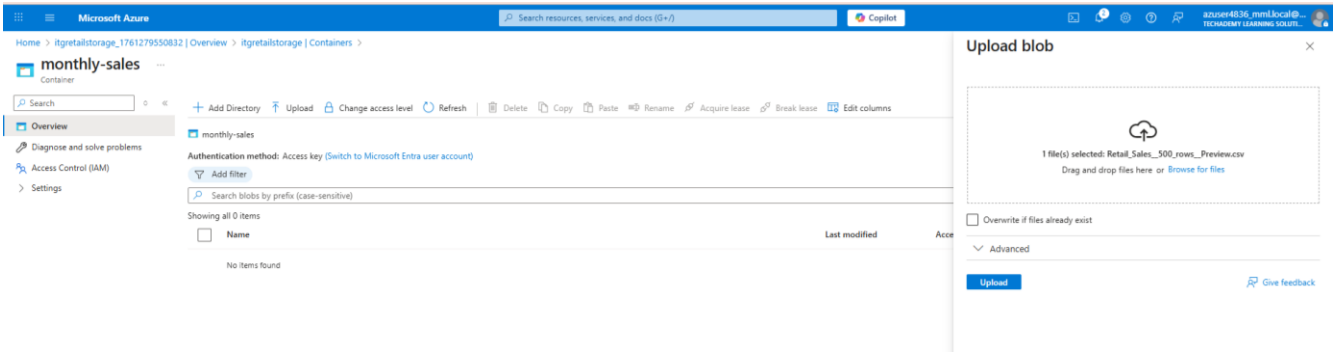
1. Inside your storage account, go to **"Containers"**.
2. Click **"+ Container"** and name it something like monthly-sales.
3. Set the access level to **Private**.

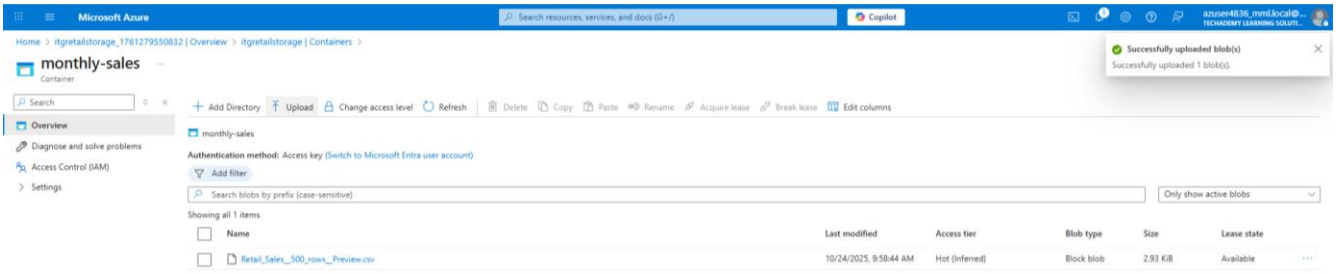


Step 3: Upload the CSV File

Option A: Azure Portal

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





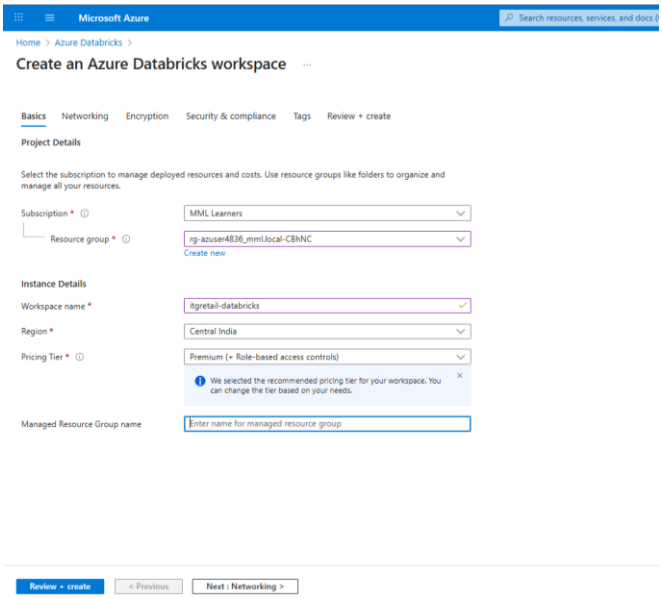
Option B: Azure Cloud Shell

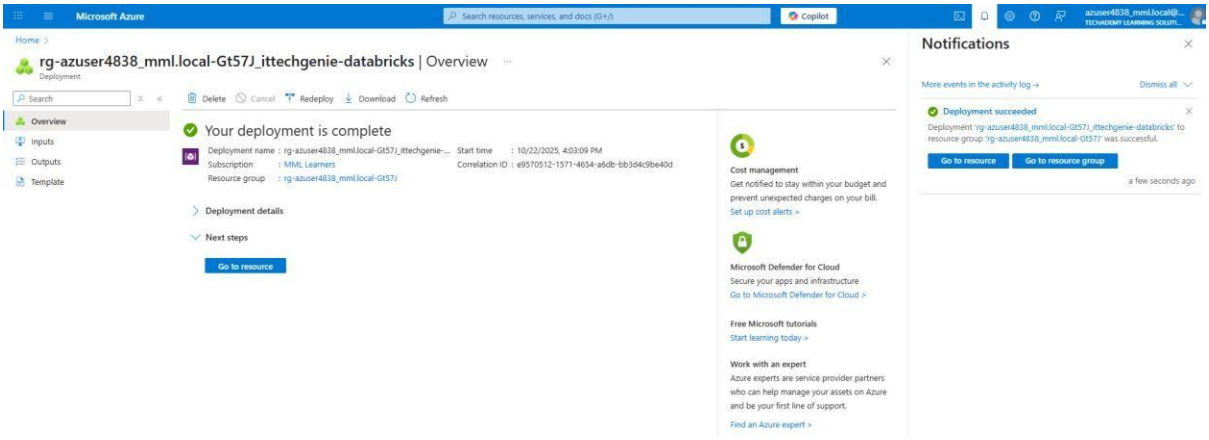
```
az storage blob upload \
--account-name itgretailstorage \
 --container-name monthly-sales \
 --name sales_october.csv \
 --file sales_october.csv \
 --auth-mode login
```

Phase 2: Provisioning Azure Databricks

Step 1: Create a Databricks Workspace

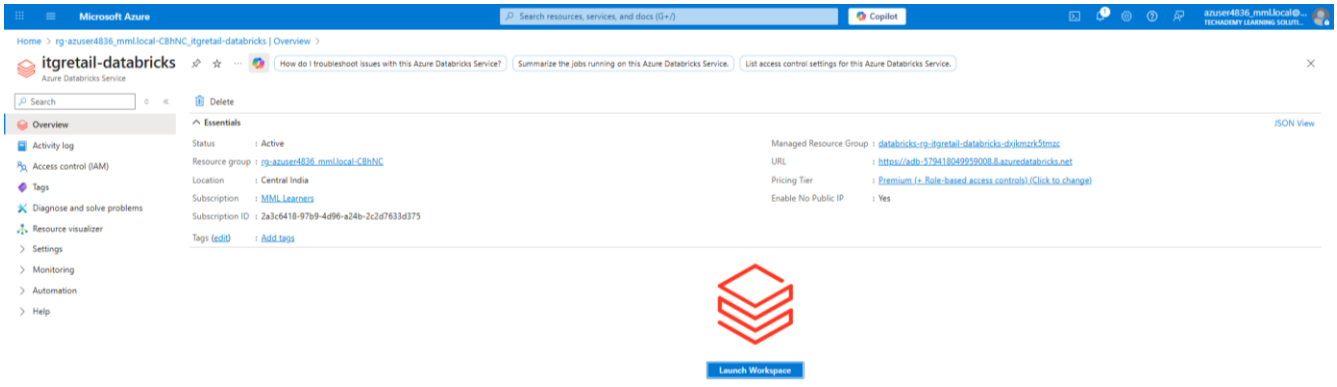
1. In Azure Portal, click **Create a resource** and search for **Azure Databricks**.
2. Set the workspace name (e.g., `itgretail-databricks`) and region (same as storage).
3. Choose the **Premium** pricing tier.
4. Click **Review + Create**, then **Create**.





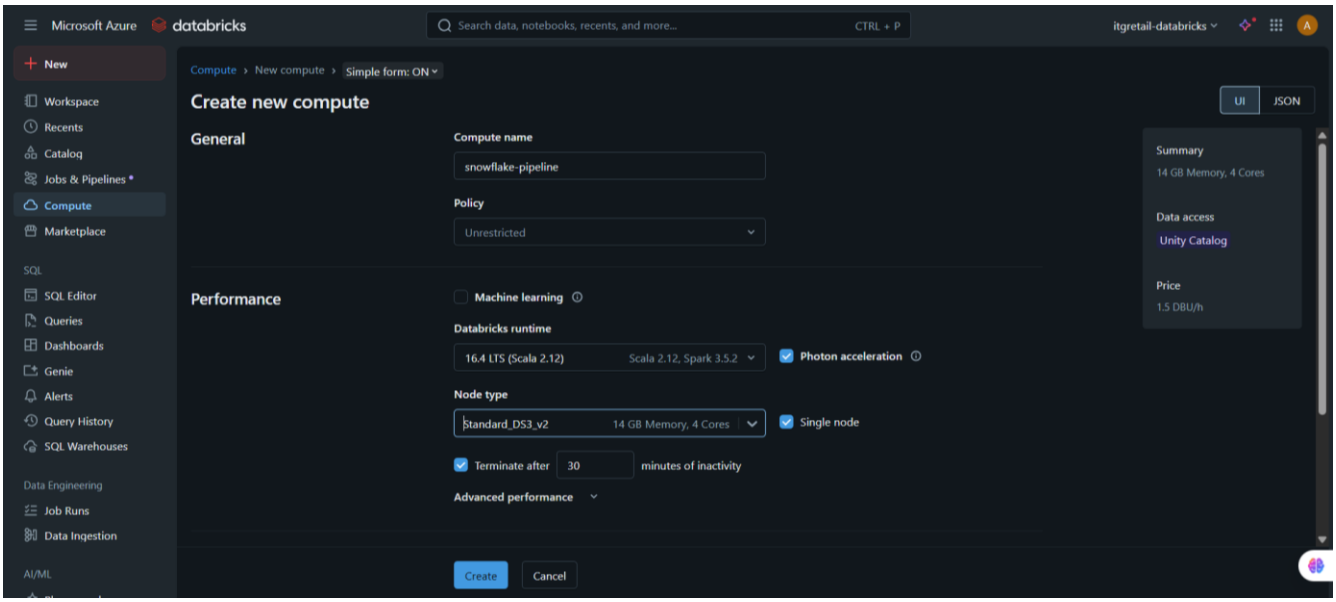
Step 2: Launch and Configure a Cluster

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

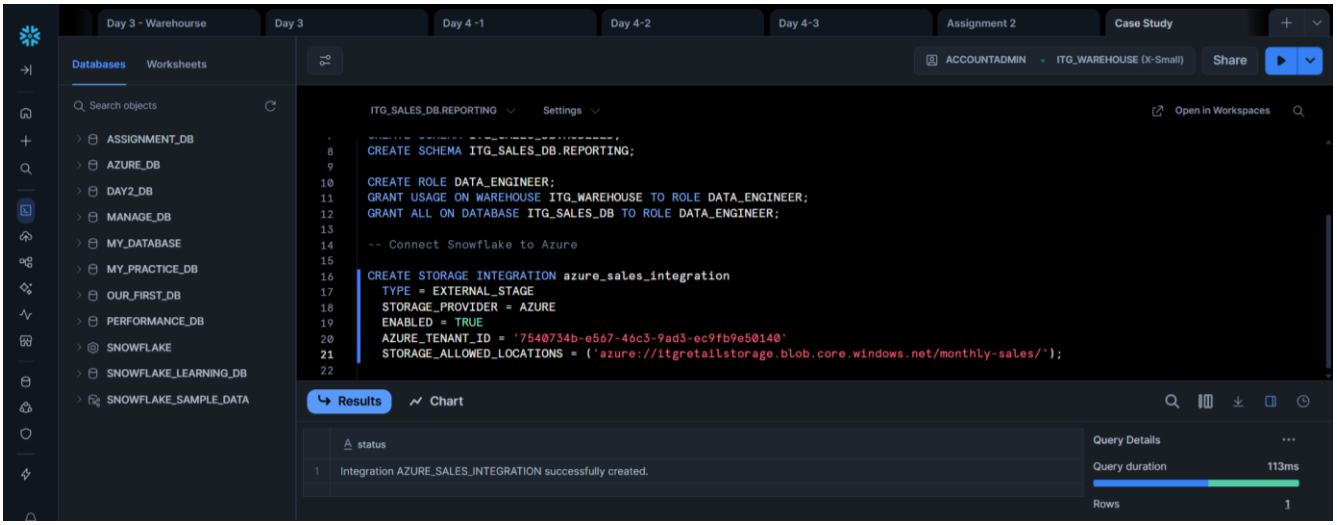
```
CREATE WAREHOUSE ITG_WAREHOUSE WITH WAREHOUSE_SIZE = XSMALL
AUTO_SUSPEND = 300 AUTO_RESUME = TRUE;

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;
```

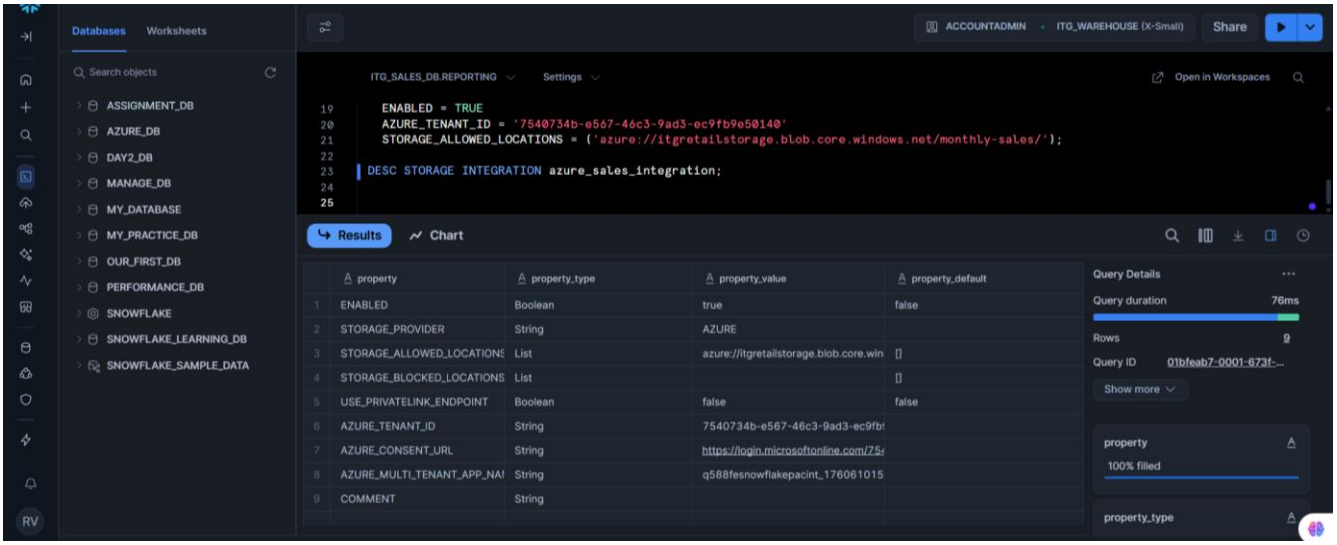
Step 2: Connect Snowflake to Azure

```
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;
```

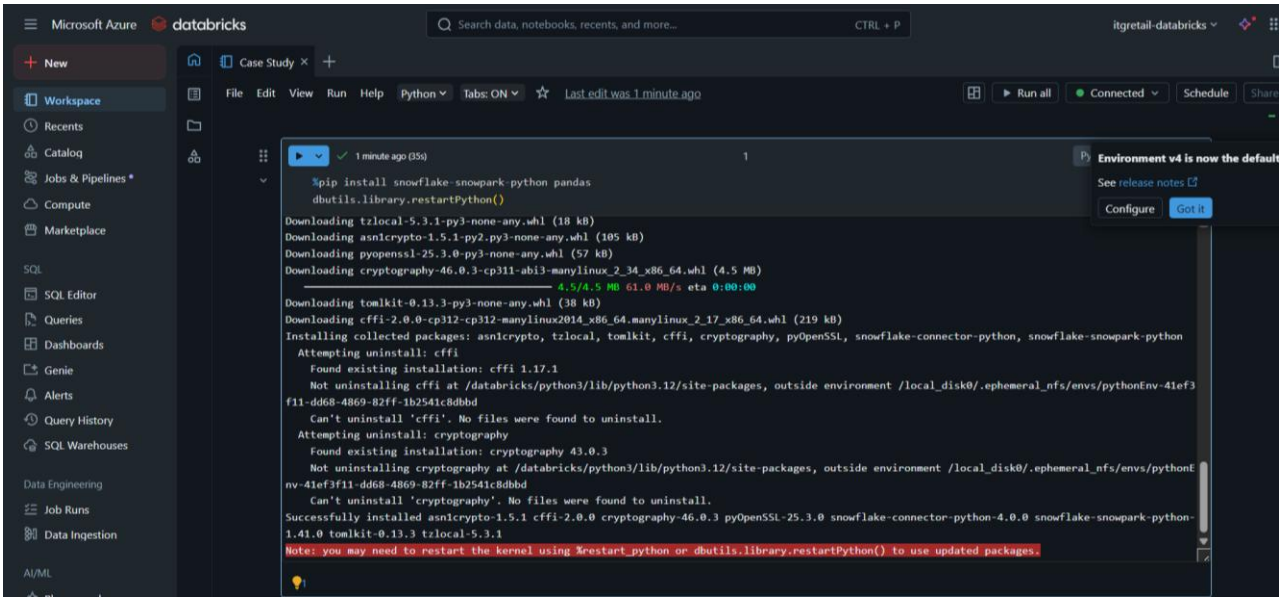


Use the AZURE_CONSENT_URL to authorize Snowflake to access your Azure storage.

Phase 4: Data Ingestion with Snowpark in Databricks

Step 1: Install Required Libraries

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

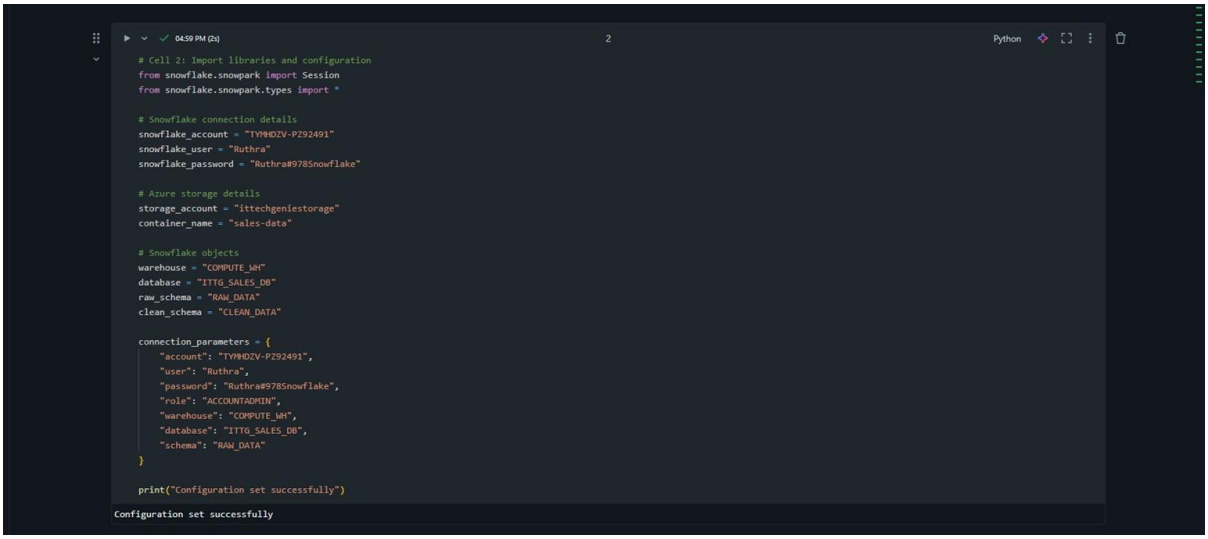


Step 2: Configure Snowflake Connection

```
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()
```



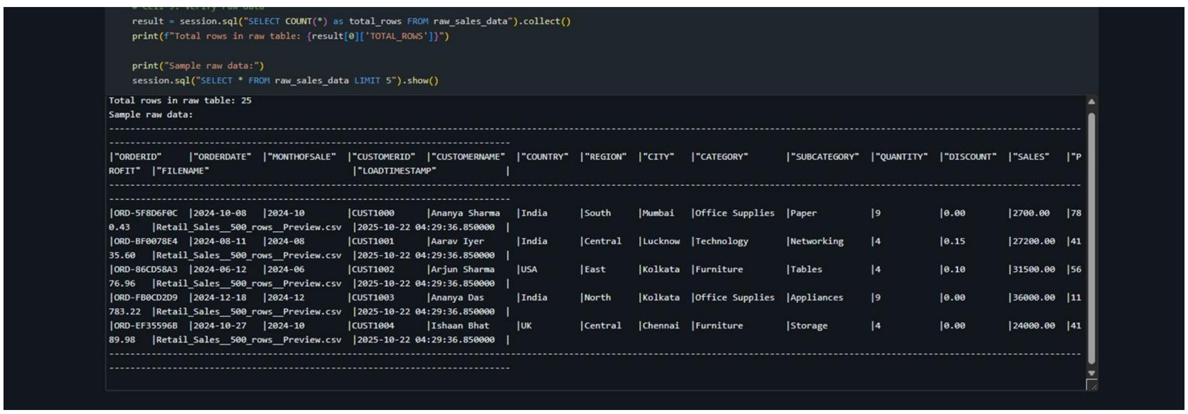
Step 3: Create Stage and File Format

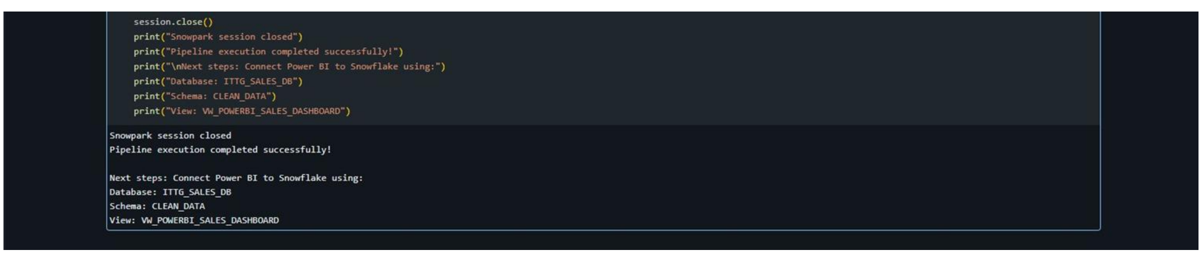
```
session.sql("""
CREATE OR REPLACE FILE FORMAT csv_format
  TYPE = 'CSV'
  FIELD_DELIMITER = ','
  SKIP_HEADER = 1
  NULL_IF = ('NULL', 'null')
  EMPTY_FIELD_AS_NULL = TRUE;
""").collect()

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()
```

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()
```

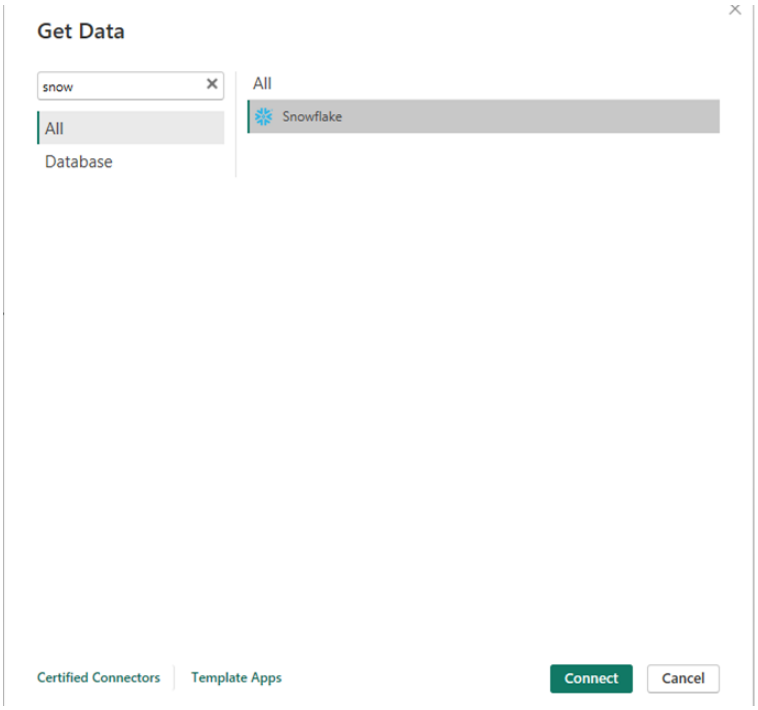




Phase 5: Power BI Reporting

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
5. Load the relevant tables or views.
6. Build visualizations like:

   · Sum of SALES by CATEGORY
   · Sum of SALES by REGION
   · Sum of SALES by REGION and CATEGORY