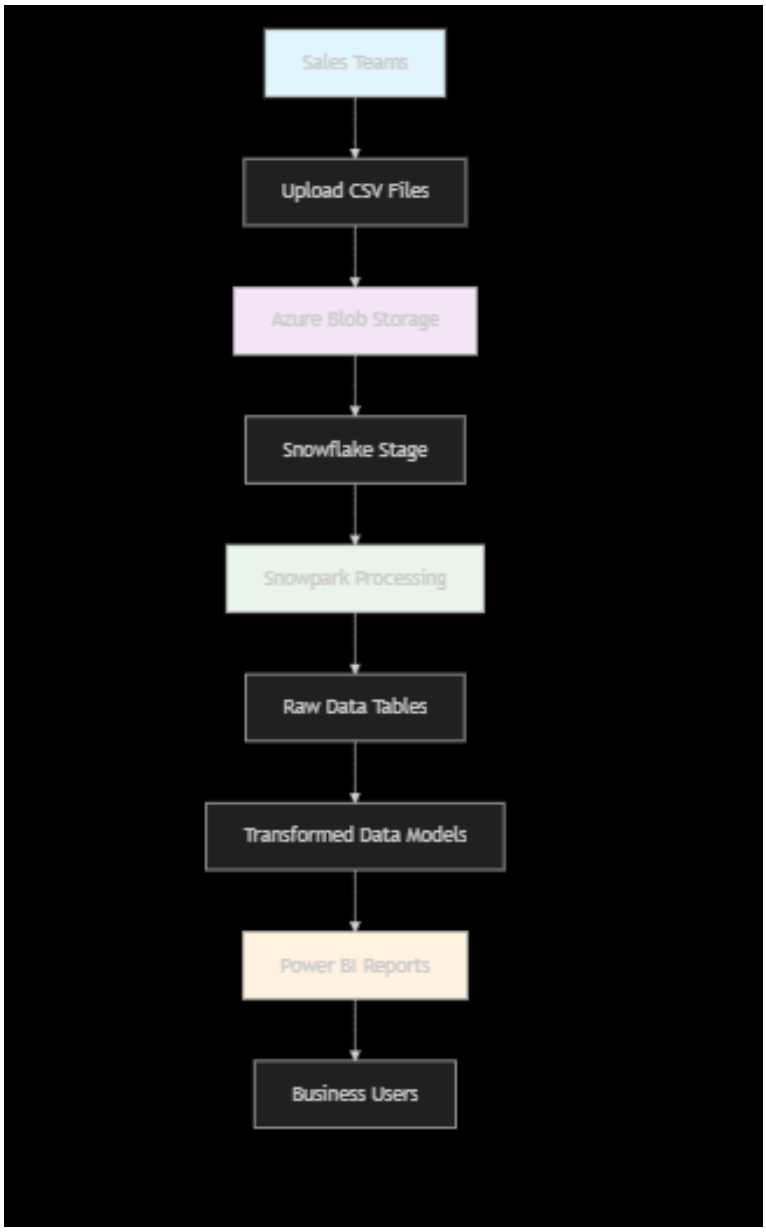
Cloud-Only Data Pipeline Implementation

Azure Blob Storage → Databricks → Snowflake → Power BI

Architecture Overview

Complete Cloud Infrastructure:

- **Data Storage**: Azure Blob Storage
- **Data Processing**: Azure Databricks
- **Data Warehouse**: Snowflake
- **Visualization**: Power BI

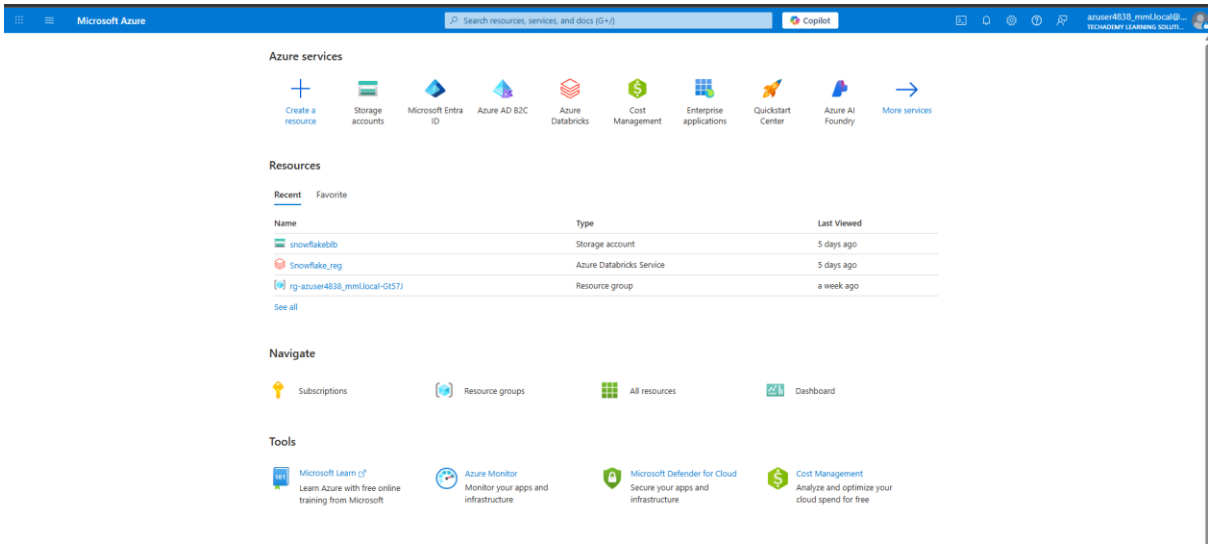


Phase 1: Azure Storage Setup

Create Azure Storage Account

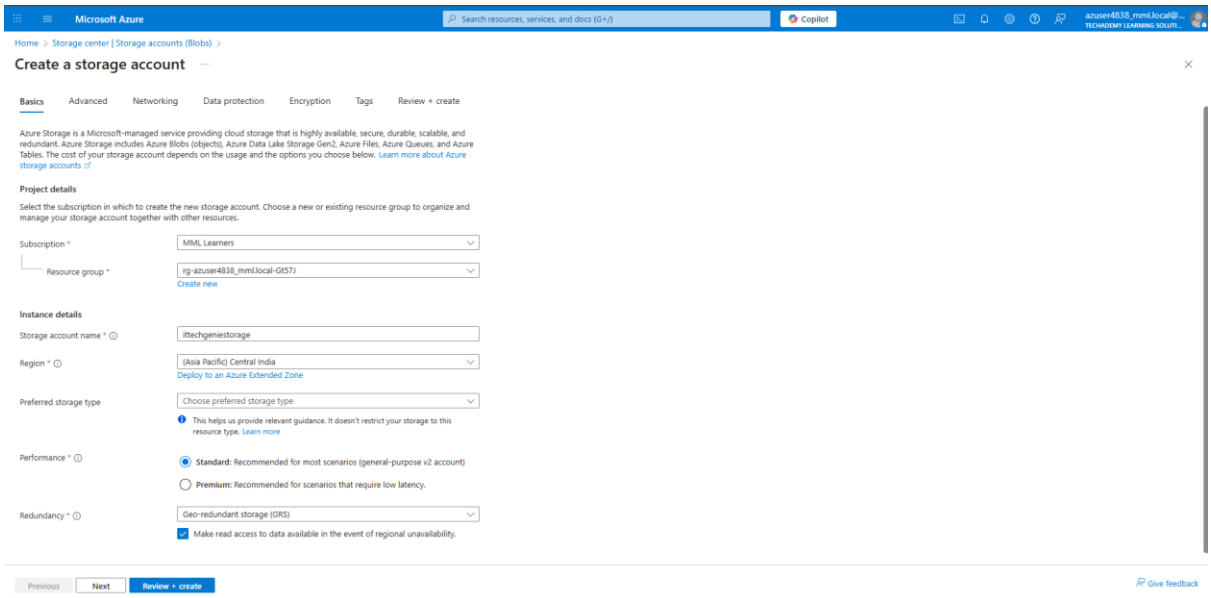
Step 1: Navigate to Azure Portal

- Go to portal.azure.com
- Sign in with your credentials



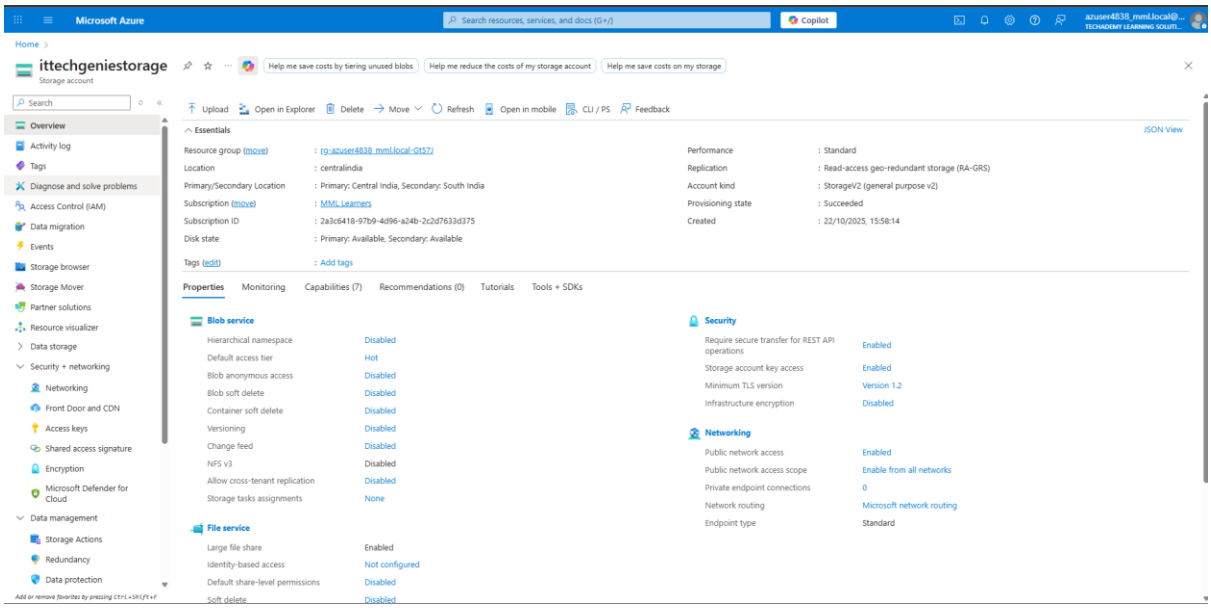
Step 2: Create Storage Account

- Click "Create a resource"
- Search for "Storage Account"
- Click "Create"



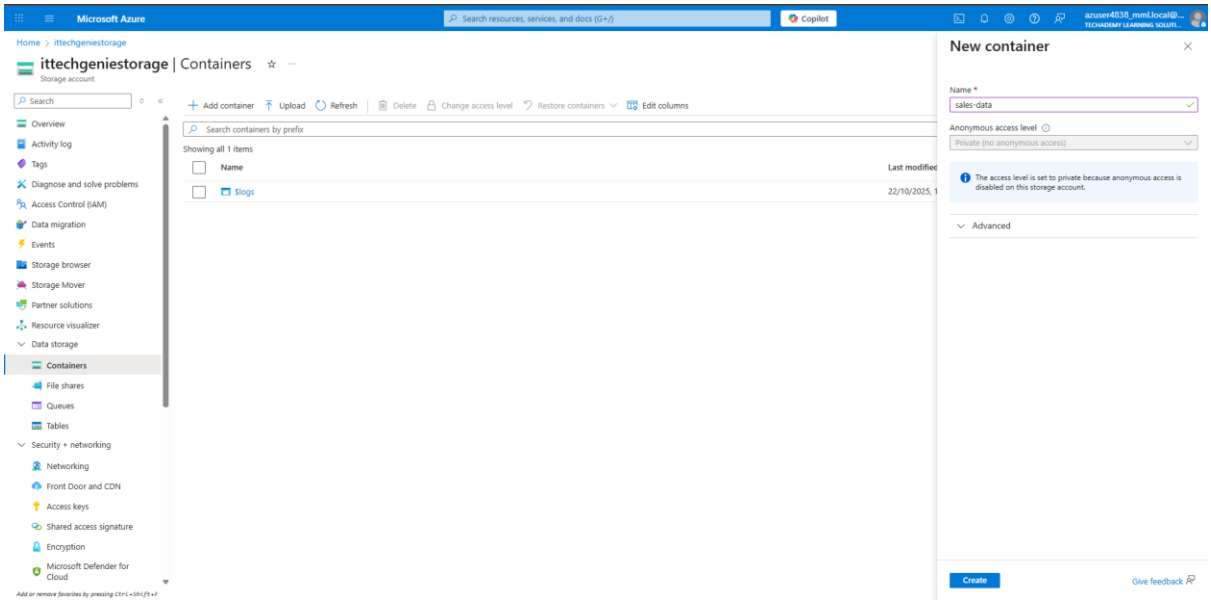
Step 3: Configure Storage Account

- **Subscription**: Select your Azure subscription
- **Resource Group**: Create new "ItTechGenie-RG"
- **Storage account name**: ittechgeniestorage
- **Region**: East US
- **Performance**: Standard
- **Redundancy**: Locally-redundant storage (LRS)



Step 4: Create Container

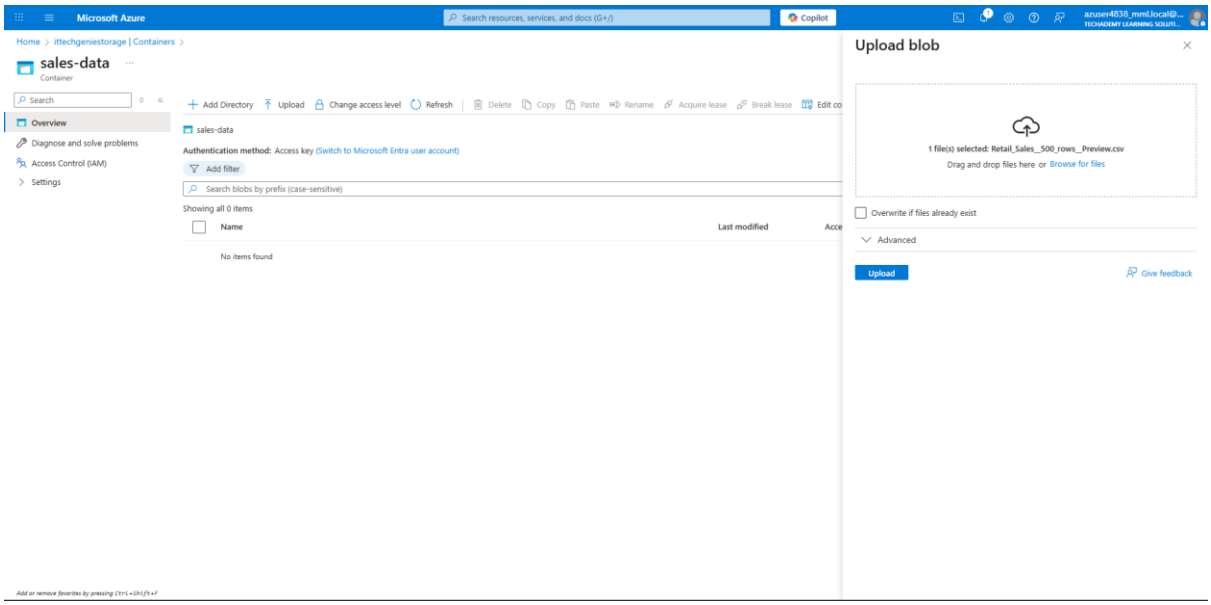
- Go to your storage account
- Navigate to "Containers" in left sidebar
- Click "+ Container"
- **Name**: sales-data
- **Public access level**: Private

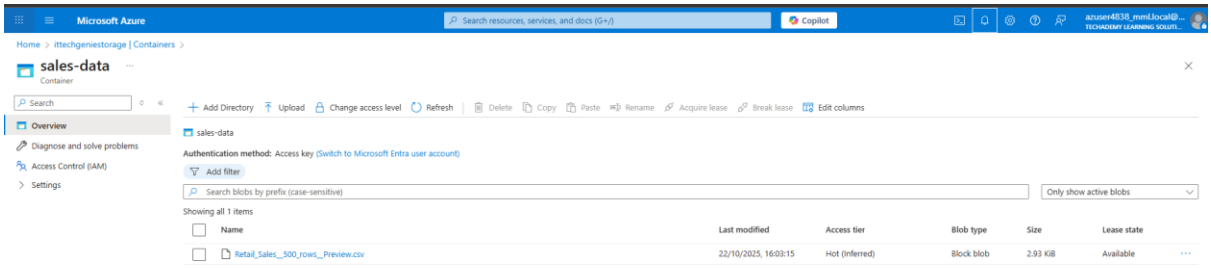


Upload Dataset to Azure Blob

Method 1: Azure Portal Upload

- Navigate to "sales-data" container
- Click "Upload" button
- Select your sales_data.csv file
- Click "Upload"





Method 2: Azure Cloud Shell

bash

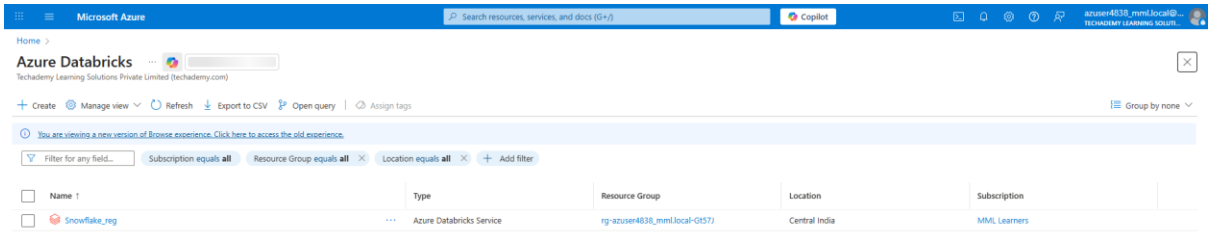
```
# Open Cloud Shell from Azure Portal (top ribbon)
# Upload your file to Cloud Shell first
az storage blob upload \
    --account-name ittechgeniestorage \
    --container-name sales-data \
    --name sales_data.csv \
    --file sales_data.csv \
    --auth-mode login
```

Phase 2: Databricks Workspace Setup

Create Databricks Workspace

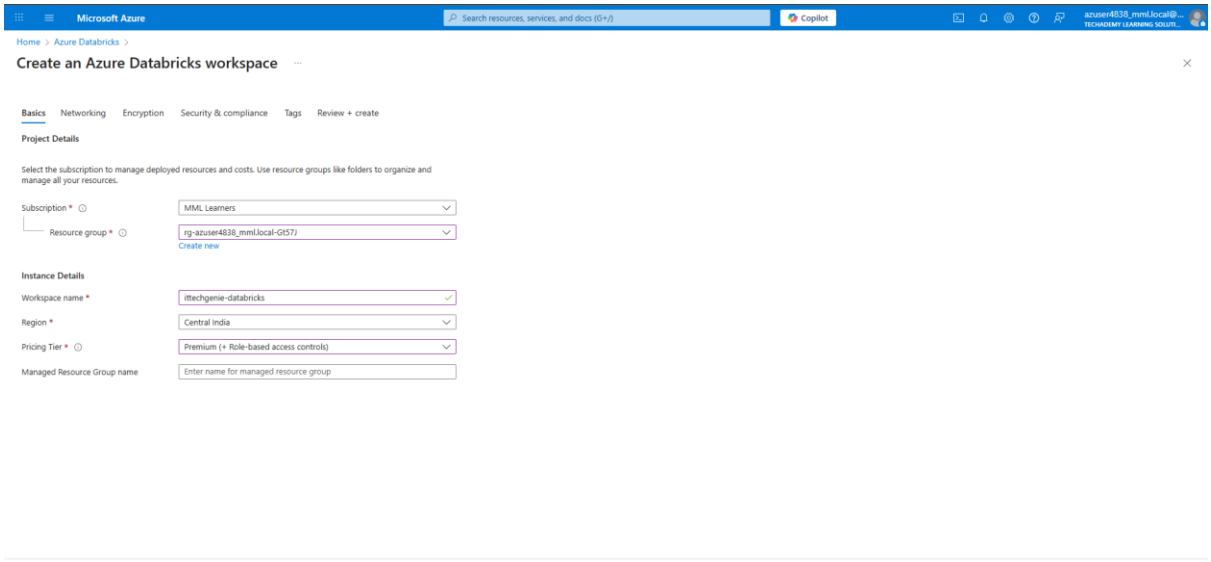
Step 1: Create Databricks Service

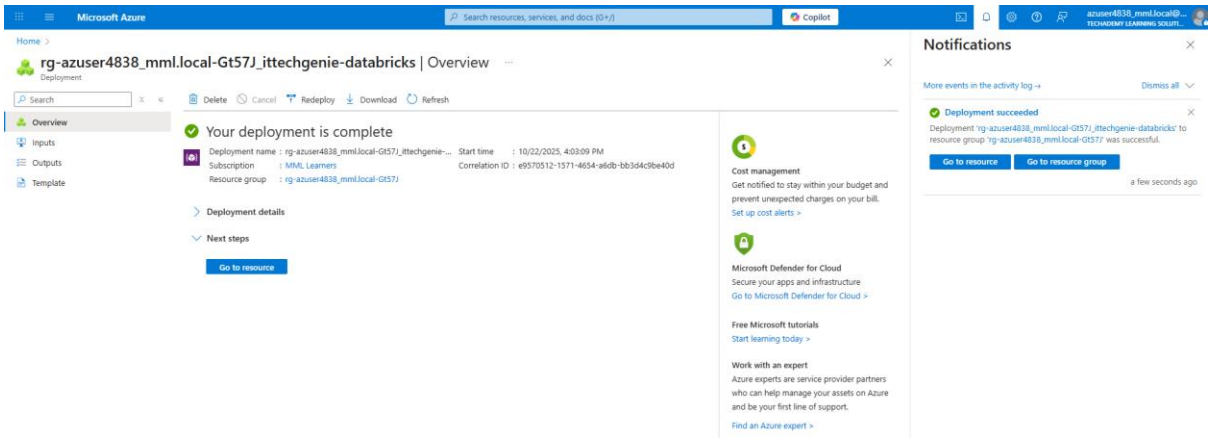
- Azure Portal → "Create a resource"
- Search for "Azure Databricks"
- Click "Create"



Step 2: Configure Workspace

- **Workspace name**: ittechgenie-databricks
- **Region**: East US (same as storage)
- **Pricing Tier**: Premium



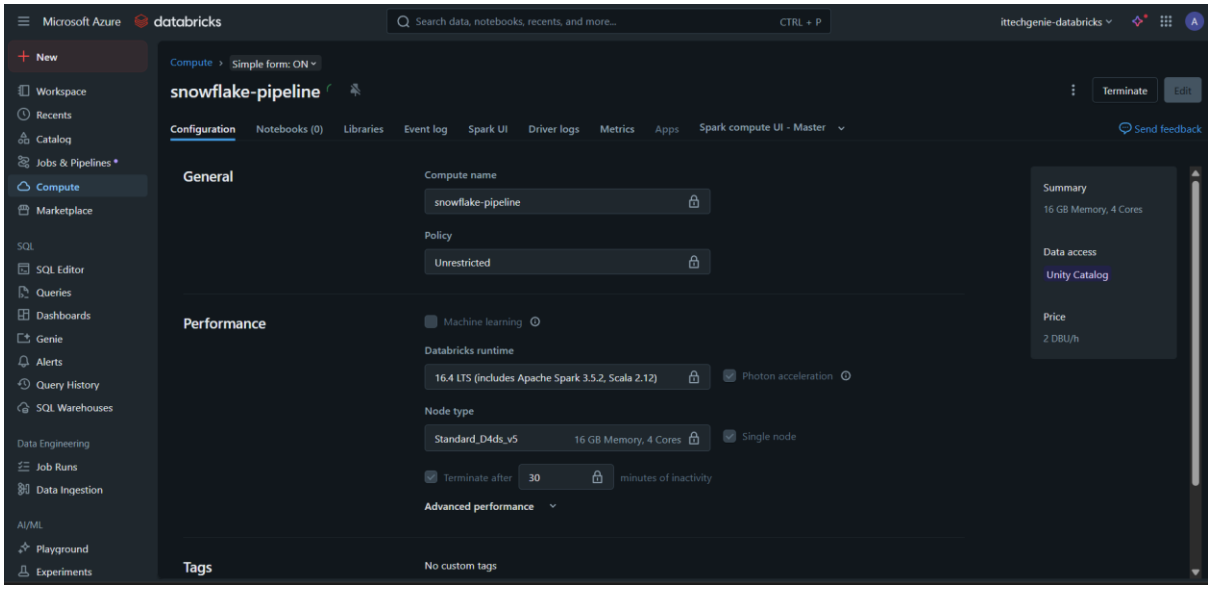


Step 3: Create Cluster

- Open Databricks Workspace
- Navigate to "Compute" in left sidebar
- Click "Create Cluster"

Step 4: Cluster Configuration

- **Cluster name**: snowflake-pipeline
- **Cluster mode**: Single Node
- **Databricks runtime version**: 12.2 LTS
- **Node type**: Standard_DS3_v2
- **Auto-termination**: 30 minutes



Phase 3: Snowflake Configuration

Snowflake Account Setup

Step 1: Create Snowflake Objects

In Snowflake web interface, run these SQL commands:

sql

```
-- Create Warehouse
CREATE WAREHOUSE ITTG_WAREHOUSE
  WAREHOUSE_SIZE = XSMALL
  AUTO_SUSPEND = 300
  AUTO_RESUME = TRUE;

-- Create Database and Schemas
CREATE DATABASE ITTG_SALES_DB;
CREATE SCHEMA ITTG_SALES_DB.RAW_DATA;
CREATE SCHEMA ITTG_SALES_DB.CLEAN_DATA;
CREATE SCHEMA ITTG_SALES_DB.ANALYTICS;

-- Create Role and Permissions
CREATE ROLE DATA_ENGINEER;
GRANT USAGE ON WAREHOUSE ITTG_WAREHOUSE TO ROLE DATA_ENGINEER;
GRANT ALL ON DATABASE ITTG_SALES_DB TO ROLE DATA_ENGINEER;
```

Step 2: Azure-Snowflake Integration

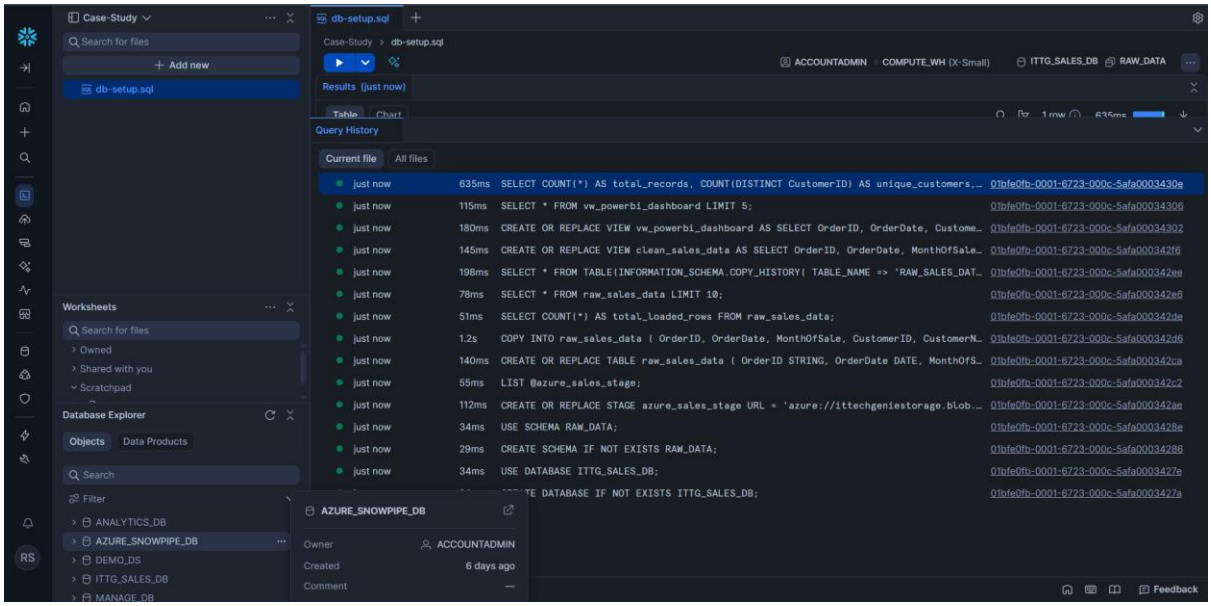
sql

```
-- Create storage integration
CREATE STORAGE INTEGRATION azure_sales_integration
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = AZURE
  ENABLED = TRUE
  AZURE_TENANT_ID = 'your-azure-tenant-id'
  STORAGE_ALLOWED_LOCATIONS = ('azure://ittechgeniestorage.blob.core.windows.net/sales-
data/');

-- Get integration details for Azure configuration
```

```
DESC STORAGE INTEGRATION azure_sales_integration;
```

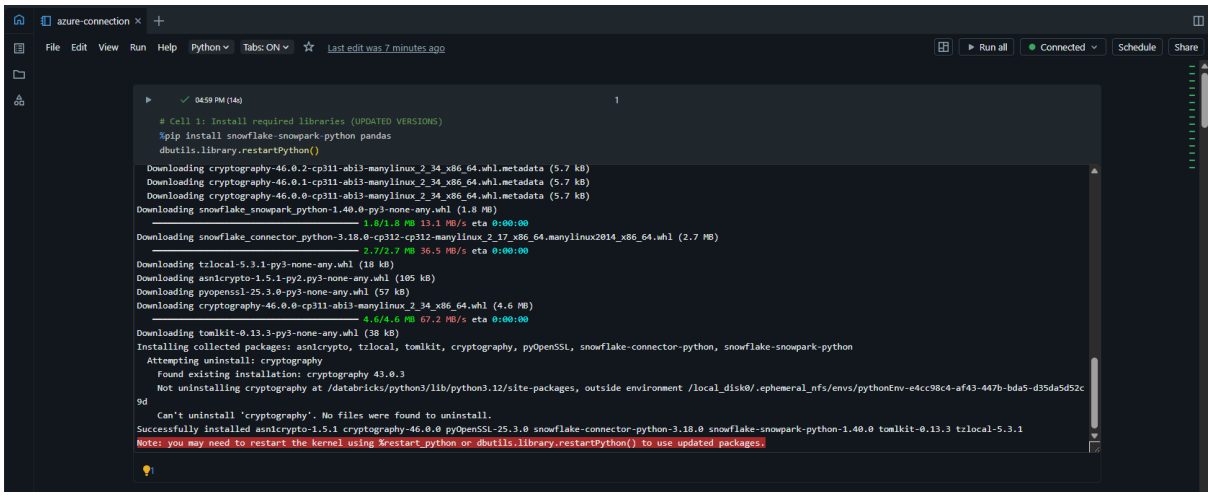
Step 3: Complete Azure Authorization

- Note the AZURE_CONSENT_URL from description output
- Open URL in browser and authenticate
- Grant Snowflake access to Azure storage



Phase 4: Databricks Pipeline Implementation

Create Databricks Notebooks



04:59 PM (2s) 2 Python

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

# Snowflake connection details
snowflake_account = "TYMHDZV-PZ92491"
snowflake_user = "Ruthra"
snowflake_password = "Ruthra#978Snowflake"

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

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

connection_parameters = {
    "account": "TYMHDZV-PZ92491",
    "user": "Ruthra",
    "password": "Ruthra#978Snowflake",
    "role": "ACCOUNTADMIN",
    "warehouse": "COMPUTE_WH",
    "database": "ITTG_SALES_DB",
    "schema": "RAW_DATA"
}

print("Configuration set successfully")
```

```
Configuration set successfully
```

04:59 PM (3s) 3

```python
# Cell 3: Create Snowpark session
session = Session.builder.configs(connection_parameters).create()
print("Snowpark session created successfully")
```

```
Snowpark session created successfully
```

04:59 PM (1s) 4 Python

```python
# Cell 4: Create database and schema if not exists
session.sql("CREATE DATABASE IF NOT EXISTS ITTG_SALES_DB").collect()
session.sql("CREATE SCHEMA IF NOT EXISTS ITTG_SALES_DB.RAW_DATA").collect()
session.sql("CREATE SCHEMA IF NOT EXISTS ITTG_SALES_DB.CLEAN_DATA").collect()
session.sql("USE DATABASE ITTG_SALES_DB").collect()
session.sql("USE SCHEMA RAW_DATA").collect()
print("Database and schema setup completed")
```

```
Database and schema setup completed
```

04:59 PM (<1s) 5 Python

```python
# Cell 5: Create file format and stage WITH SAS TOKEN
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=racwdl&st=2025-10-22T10:47:09Z&se=2025-10-23T19:02:09Z&spr=https&sv=2024-11-04&sr=c&sig=hEF7601nEZP%2Byvbuk9F2FVtAou%2F3%2BoDvfC3fNQ5fLbs%3D'
    )
    FILE_FORMAT = csv_sales_format;
""").collect()

print("File format and stage created successfully")
```

```
File format and stage created successfully
```

+ Code + Text Assistant

04:59 PM (<1s) 6

```python
# Cell 6: Test stage connection
try:
    result = session.sql("LIST @azure_sales_stage").collect()
    print("Stage connection successful! Files found:")
    for row in result:
        print(f" - {row['name']}")
except Exception as e:
    print(f"Error listing files: {str(e)}")
```

```
Stage connection successful! Files found:
 - azure://ittechgeniestorage.blob.core.windows.net/sales-data/Retail_Sales__500_rows__Preview.csv
```

04:59 PM (1s) 7 Python

```python
# Cell 7: Create raw table schema matching your CSV
session.sql("""
CREATE OR REPLACE TABLE raw_sales_data (
    OrderID STRING,
    OrderDate DATE,
    MonthOfSale STRING,
    CustomerID STRING,
    CustomerName STRING,
    Country STRING,
    Region STRING,
    City STRING,
    Category STRING,
    Subcategory STRING,
    Quantity INTEGER,
    Discount NUMBER(10,2),
    Sales NUMBER(10,2),
    Profit NUMBER(10,2),
    FileName STRING,
    LoadTimestamp TIMESTAMP_NTZ DEFAULT CURRENT_TIMESTAMP()
);
""").collect()

print("Raw sales table created successfully")
```

```
Raw sales table created successfully
```

04:59 PM (2s) 8

```python
# Cell 8: Ingest data from Azure to Snowflake
copy_result = 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_sales_stage/Retail_Sales__500_rows__Preview.csv
)
FILE_FORMAT = (FORMAT_NAME = csv_sales_format)
ON_ERROR = 'CONTINUE';
""").collect()

print(f"Data ingestion completed: {copy_result[0]['rows_loaded']} rows loaded")
```

```
Data ingestion completed: 25 rows loaded
```

04:59 PM (<1s) 9

```python
# Cell 9: Verify raw data
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"   |"P
ROFIT"  |"FILENAME"                         |"LOADTIMESTAMP"              |
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------
|ORD-5F8D6F0C  |2024-10-08   |2024-10        |CUST1000      |Ananya Sharma   |India      |South     |Mumbai   |Office Supplies  |Paper          |9           |0.00        |2700.00   |78
0.43    |Retail_Sales__500_rows__Preview.csv  |2025-10-22 04:29:36.850000  |
|ORD-BF0078E4  |2024-08-11   |2024-08        |CUST1001      |Aarav Iyer      |India      |Central   |Lucknow  |Technology       |Networking     |4           |0.15        |27200.00  |41
35.60   |Retail_Sales__500_rows__Preview.csv  |2025-10-22 04:29:36.850000  |
|ORD-86CD58A3  |2024-06-12   |2024-06        |CUST1002      |Arjun Sharma    |USA        |East      |Kolkata  |Furniture        |Tables         |4           |0.10        |31500.00  |56
76.96   |Retail_Sales__500_rows__Preview.csv  |2025-10-22 04:29:36.850000  |
|ORD-FB0CD2D9  |2024-12-18   |2024-12        |CUST1003      |Ananya Das      |India      |North     |Kolkata  |Office Supplies  |Appliances     |9           |0.00        |36000.00  |11
783.22  |Retail_Sales__500_rows__Preview.csv  |2025-10-22 04:29:36.850000  |
|ORD-EF35596B  |2024-10-27   |2024-10        |CUST1004      |Ishaan Bhat     |UK         |Central   |Chennai  |Furniture        |Storage        |4           |0.00        |24000.00  |41
89.98   |Retail_Sales__500_rows__Preview.csv  |2025-10-22 04:29:36.850000  |
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------
```

04:59 PM (1s) 10 Python

```python
# Cell 10: Switch to clean schema and create clean data table
session.sql("USE SCHEMA CLEAN_DATA").collect()

session.sql("""
CREATE OR REPLACE TABLE clean_sales_data AS
SELECT
    OrderID,
    OrderDate,
    MonthOfSale,
    CustomerID,
    CustomerName,
    Country,
    Region,
    City,
    Category,
    Subcategory,
    Quantity,
    Discount,
    Sales,
    Profit,
    -- Data validation and calculations
    CASE WHEN Sales != Quantity * (Sales/NULLIF(Quantity,0)) THEN Sales ELSE Sales END AS ValidatedSales,
    -- Date parts for analysis
    YEAR(OrderDate) AS OrderYear,
    MONTH(OrderDate) AS OrderMonth,
    QUARTER(OrderDate) AS OrderQuarter,
    -- Business metrics
    Sales * Discount AS DiscountAmount,
    Profit / NULLIF(Sales, 0) AS ProfitMargin,
    LoadTimestamp
FROM ITTG_SALES_DB.RAW_DATA.raw_sales_data
WHERE OrderDate IS NOT NULL AND Sales > 0;
""").collect()

print("Clean sales data table created")
```

```
Clean sales data table created
```

04:59 PM (<1s) 11 Python

```python
# Cell 11: Create aggregated views for Power BI
session.sql("""
CREATE OR REPLACE VIEW sales_summary_monthly AS
SELECT
    Region,
    Category,
    OrderYear,
    OrderMonth,
    COUNT(*) AS TotalOrders,
    SUM(Sales) AS TotalSales,
    SUM(Profit) AS TotalProfit,
    AVG(Sales) AS AvgOrderValue,
    SUM(Quantity) AS TotalQuantity,
    COUNT(DISTINCT CustomerID) AS UniqueCustomers
FROM clean_sales_data
GROUP BY Region, Category, OrderYear, OrderMonth
ORDER BY OrderYear, OrderMonth, Region;
""").collect()

print("Aggregated views created successfully")
```

```
Aggregated views created successfully
```

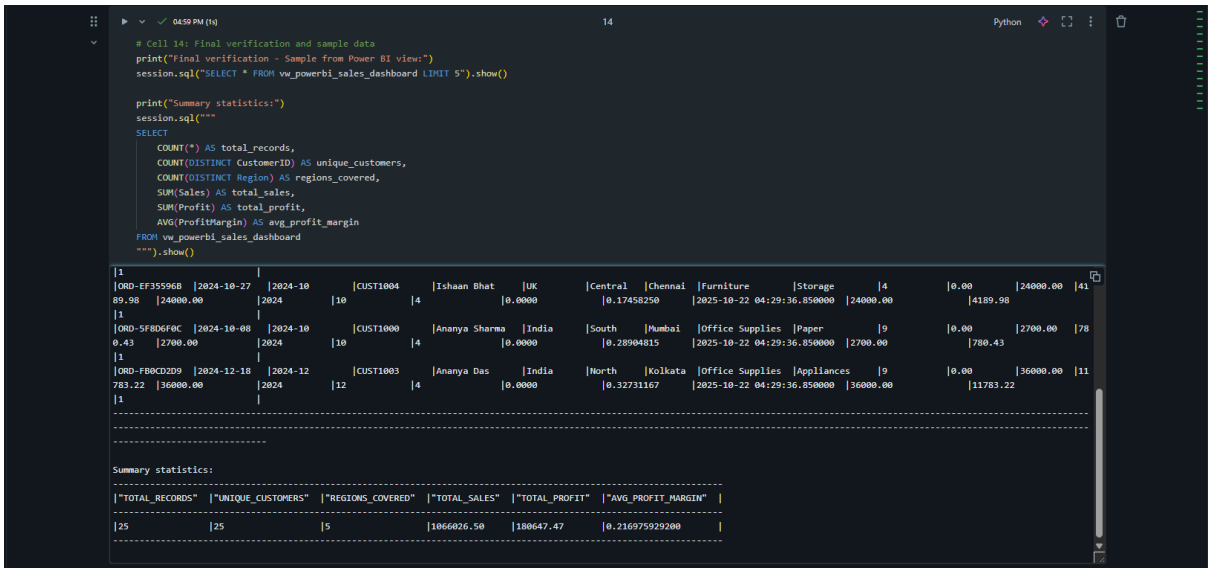
04:59 PM (<1s) 12

```python
# Cell 12: Create Power BI optimized view
session.sql("""
CREATE OR REPLACE VIEW vw_powerbi_sales_dashboard AS
SELECT
    cs.*,
    sm.TotalSales AS RegionMonthlySales,
    sm.TotalProfit AS RegionMonthlyProfit,
    sm.UniqueCustomers AS RegionMonthlyCustomers
FROM clean_sales_data cs
LEFT JOIN sales_summary_monthly sm
    ON cs.Region = sm.Region
    AND cs.Category = sm.Category
    AND cs.OrderYear = sm.OrderYear
    AND cs.OrderMonth = sm.OrderMonth;
""").collect()

print("Power BI view created successfully")
```

```
Power BI view created successfully
```

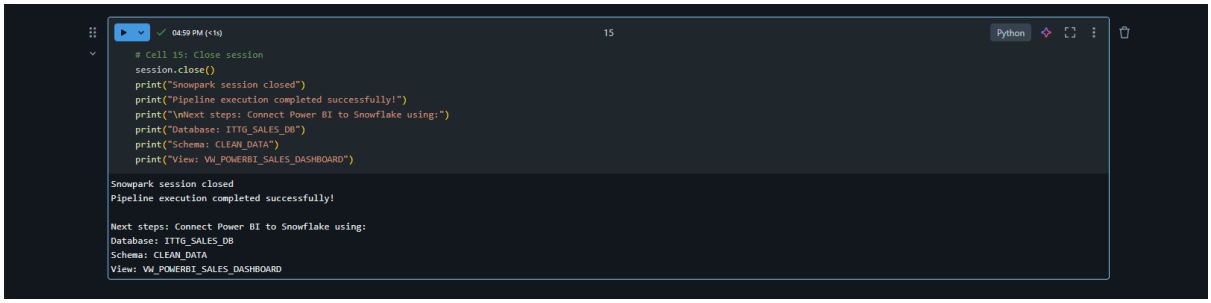
```
# Cell 13: Data quality checks
checks = [
    "SELECT COUNT(*) AS total_rows FROM raw_sales_data",
    "SELECT COUNT(*) AS clean_rows FROM clean_sales_data",
    "SELECT COUNT(DISTINCT Region) AS region_count FROM clean_sales_data",
    "SELECT MIN(OrderDate) AS earliest_date, MAX(OrderDate) AS latest_date FROM clean_sales_data",
    "SELECT SUM(Sales) AS total_sales, SUM(Profit) AS total_profit FROM clean_sales_data"
]

print("Data Quality Check Results:")
for check in checks:
    try:
        result = session.sql(check).collect()
        print(f"{check}: {result[0]}")
    except Exception as e:
        print(f"Error in check {check}: {str(e)}")
```

```
Data Quality Check Results:
Error in check SELECT COUNT(*) AS total_rows FROM raw_sales_data: (1304): 01bfe111-0001-6723-000c-5afa000343a2: 002003 (42S02): SQL compilation error:
Object 'RAW_SALES_DATA' does not exist or not authorized.
SELECT COUNT(*) AS clean_rows FROM clean_sales_data: Row(CLEAN_ROWS=25)
SELECT COUNT(DISTINCT Region) AS region_count FROM clean_sales_data: Row(REGION_COUNT=5)
SELECT MIN(OrderDate) AS earliest_date, MAX(OrderDate) AS latest_date FROM clean_sales_data: Row(EARLIEST_DATE=datetime.date(2024, 2, 27), LATEST_DATE=datetime.date(2025, 9, 15))
SELECT SUM(Sales) AS total_sales, SUM(Profit) AS total_profit FROM clean_sales_data: Row(TOTAL_SALES=Decimal('1066026.50'), TOTAL_PROFIT=Decimal('180647.47'))
```



```
# Cell 14: Final verification and sample data
print("Final verification - Sample from Power BI view:")
session.sql("SELECT * FROM vw_powerbi_sales_dashboard LIMIT 5").show()

print("Summary statistics:")
session.sql("""
SELECT
    COUNT(*) AS total_records,
    COUNT(DISTINCT CustomerID) AS unique_customers,
    COUNT(DISTINCT Region) AS regions_covered,
    SUM(Sales) AS total_sales,
    SUM(Profit) AS total_profit,
    AVG(ProfitMargin) AS avg_profit_margin
FROM vw_powerbi_sales_dashboard
""").show()
```

```
|1                        |
|ORD-EF35596B  |2024-10-27  |2024-10       |CUST1004      |Ishaan Bhat     |UK          |Central   |Chennai  |Furniture       |Storage       |4          |0.00       |24000.00  |41
89.98   |24000.00          |2024          |10            |4               |0.0000          |0.17458250      |2025-10-22 04:29:36.850000  |24000.00            |4189.98
|1                        |
|ORD-5F8D6F0C  |2024-10-08  |2024-10       |CUST1000      |Ananya Sharma   |India       |South     |Mumbai   |Office Supplies |Paper         |9          |0.00       |2700.00   |78
0.43    |2700.00           |2024          |10            |4               |0.0000          |0.28904815      |2025-10-22 04:29:36.850000  |2700.00             |780.43
|1                        |
|ORD-FB0CD2D9  |2024-12-18  |2024-12       |CUST1003      |Ananya Das      |India       |North     |Kolkata  |Office Supplies |Appliances    |9          |0.00       |36000.00  |11
783.22  |36000.00          |2024          |12            |4               |0.0000          |0.32731167      |2025-10-22 04:29:36.850000  |36000.00            |11783.22
|1                        |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------

Summary statistics:
-------------------------------------------------------------------------------------------
|"TOTAL_RECORDS"  |"UNIQUE_CUSTOMERS"  |"REGIONS_COVERED"  |"TOTAL_SALES"  |"TOTAL_PROFIT"  |"AVG_PROFIT_MARGIN"  |
-------------------------------------------------------------------------------------------
|25               |25                  |5                  |1066026.50     |180647.47       |0.216975929200       |
-------------------------------------------------------------------------------------------
```



```
# Cell 15: Close session
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

Connect Power BI to Snowflake

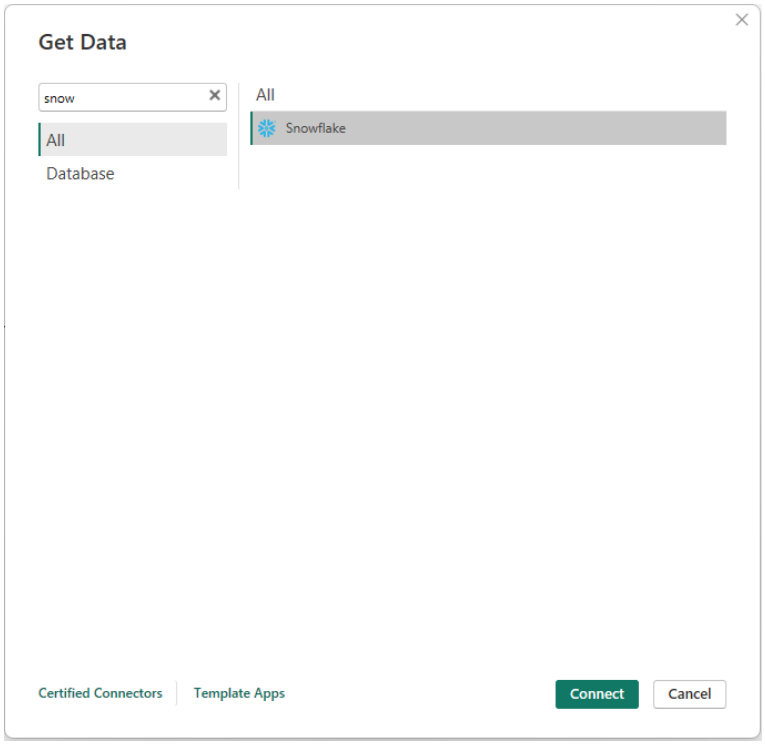
Step 1: Open Power BI Desktop

- Launch Power BI Desktop on your local machine

- This is the ONLY step that happens locally

Step 2: Snowflake Connection

- Click "Get Data"
- Search for "Snowflake"
- Click "Connect"

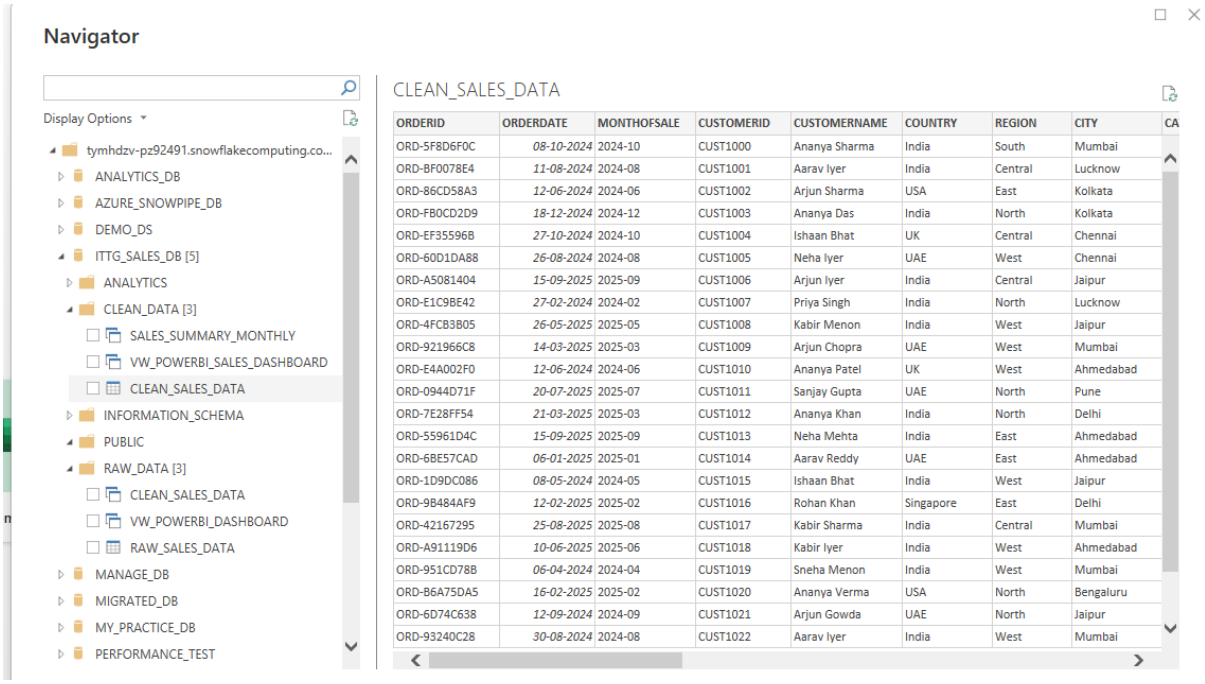


Step 3: Connection Details

- **Server**: your_account.snowflakecomputing.com
- **Warehouse**: ITTG_WAREHOUSE
- **Database**: ITTG_SALES_DB
- **Schema**: CLEAN_DATA

Step 4: Import Data

- Select "Import" connectivity mode
- Choose the views: VW_POWERBI_SALES_DASHBOARD and SALES_SUMMARY_MONTHLY
- Click "Load"



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-B6A75DA5	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	

Create Power BI Report

Recommended Visualizations:

Page 1: Sales Overview

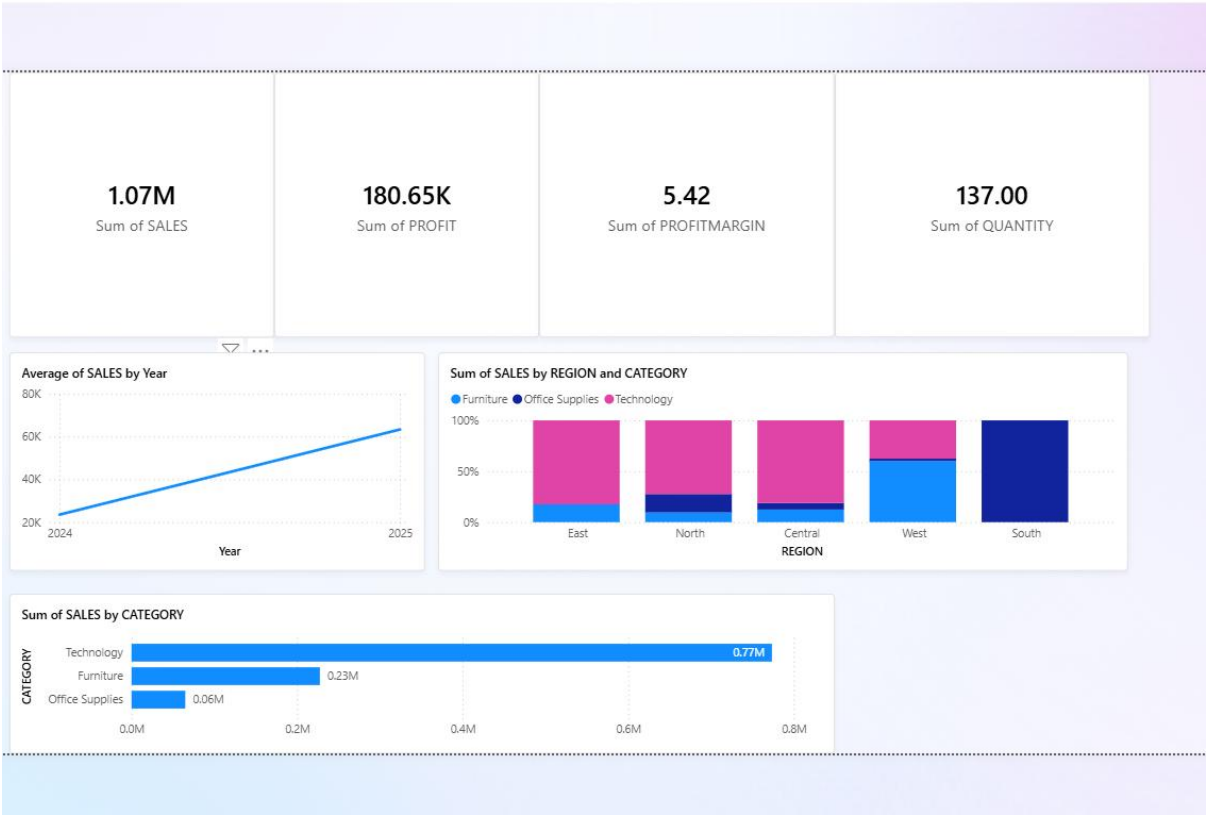
- **Total Sales** (Card visual)
- **Total Profit** (Card visual)
- **Sales by Month** (Line chart)
- **Sales by Region** (Stacked column chart)
- **Top Categories** (Bar chart)

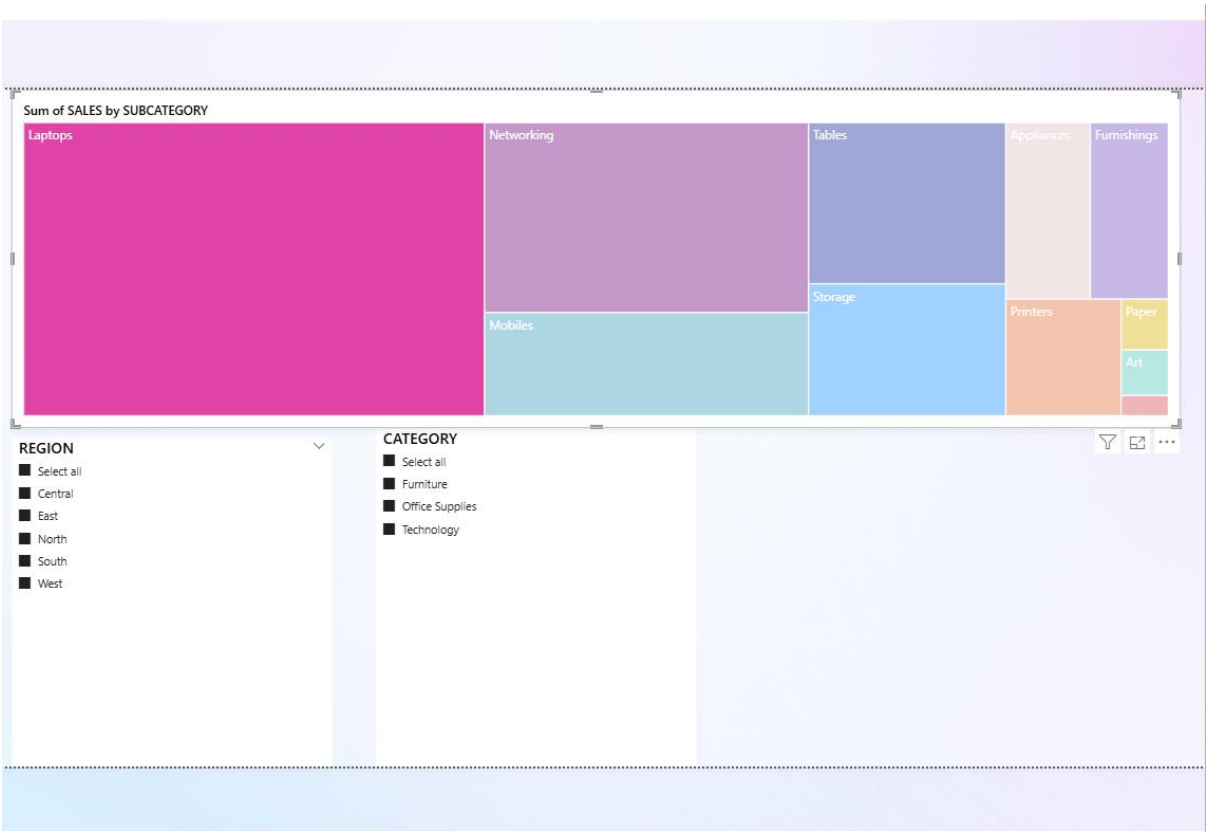
Page 2: Regional Performance

- **Sales by Region** (Map visual)
- **Profit Margin by Category** (Matrix visual)
- **Customer Distribution** (Pie chart)
- **Monthly Trends by Region** (Line chart)

Page 3: Product Analysis

- **Sales by Subcategory** (Treemap)
- **Quantity vs Profit** (Scatter chart)
- **Discount Impact** (Line and clustered column chart)


1.07M
Sum of SALES
180.65K
Sum of PROFIT
5.42
Sum of PROFITMARGIN
137.00
Sum of QUANTITY
Average of SALES by Year
80K
60K
40K
20K
2024
2025
Year
Sum of SALES by REGION and CATEGORY
Furniture
Office Supplies
Technology
100%
50%
0%
East
North
Central
West
South
REGION
Sum of SALES by CATEGORY
CATEGORY
Technology
Furniture
Office Supplies
0.77M
0.23M
0.06M
0.0M
0.2M
0.4M
0.6M
0.8M


Sum of SALES by SUBCATEGORY
Laptops
Networking
Tables
Appliances
Furnishings
Storage
Printers
Paper
Art
Mobiles
REGION
Select all
Central
East
North
South
West
CATEGORY
Select all
Furniture
Office Supplies
Technology