In [2]:
```
os.getcwd()
```

Out[2]:
```
'C:\\Users\\Anand Jha'
```

In [1]:
```
import boto3
import botocore
from botocore.config import Config
import getpass
import snowflake.connector
import pandas as pd
import os
from io import StringIO
import csv
```

In [14]:
```
# Establish the connection
conn = snowflake.connector.connect(
    account= 'fybxjxj-ot90647',
    user='ANALYTICSWITHANAND',
    password = getpass.getpass('Your Snowflake Password: '),
    warehouse='DEMO_WAREHOUSE',
    database='DEMO_DATABASE',
    schema='DEMO_SCHEMA',
    role='ACCOUNTADMIN'
)
```

```
Your Snowflake Password: ·····
```

In [15]:
```
# Test the connection
cursor = conn.cursor()
cursor.execute("SELECT CURRENT_VERSION()")
print(cursor.fetchone())
```

```
('9.1.0',)
```

In [17]:
```
## Set up AWS credentials manually (only for testing)
aws_access_key_id = 'AKIAZUFV73CQJNCODF4O'
aws_secret_access_key = 'JhOqgC6uMUXMvRj9gCUKFI+5rEnr0LBQ9j0mai+0'
region_name = 'ap-southeast-1'  # Replace with your region
```

In [18]:
```
# Create a session using the manual credentials
session = boto3.Session(
    aws_access_key_id=aws_access_key_id,
    aws_secret_access_key=aws_secret_access_key,
    region_name=region_name
)
```

In [19]:
```
# Create an S3 client
s3 = session.client('s3')
```

In [20]:
```
# Now you can use the S3 client to perform operations to list all buckets
response = s3.list_buckets()
print(response)
```

{'ResponseMetadata': {'RequestId': '68R9GP3JF0Y82GP8', 'HostId': '27iHRxDb3nXOuL+5J8AnUv60yC2jT0fjl5NBNxXcJEaTe8wYzPu9I84KaQE5RNQadBNvq9Pz+Xs=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': '27iHRxDb3nXOuL+5J8AnUv60yC2jT0fjl5NBNxXcJEaTe8wYzPu9I84KaQE5RNQadBNvq9Pz+Xs=', 'x-amz-request-id': '68R9GP3JF0Y82GP8', 'date': 'Mon, 20 Jan 2025 15:53:08 GMT', 'content-type': 'application/xml', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 0}, 'Buckets': [{'Name': 'aj-calender', 'CreationDate': datetime.datetime(2024, 7, 10, 21, 17, 33, tzinfo=tzutc())}, {'Name': 'aj-iotv2', 'CreationDate': datetime.datetime(2024, 2, 11, 21, 56, 11, tzinfo=tzutc())}, {'Name': 'aj-northwind-data', 'CreationDate': datetime.datetime(2023, 10, 28, 20, 7, 4, tzinfo=tzutc())}, {'Name': 'aws-glue-assets-661806635168-us-east-1', 'CreationDate': datetime.datetime(2024, 12, 21, 16, 11, 3, tzinfo=tzutc())}, {'Name': 'czec-banking', 'CreationDate': datetime.datetime(2024, 11, 9, 11, 34, 19, tzinfo=tzutc())}, {'Name': 'mat-pi', 'CreationDate': datetime.datetime(2024, 7, 28, 16, 9, 53, tzinfo=tzutc())}, {'Name': 'matillionclass', 'CreationDate': datetime.datetime(2024, 12, 4, 9, 4, 48, tzinfo=tzutc())}, {'Name': 'matillionprac', 'CreationDate': datetime.datetime(2024, 7, 23, 11, 8, 20, tzinfo=tzutc())}, {'Name': 'mybuckethello', 'CreationDate': datetime.datetime(2024, 12, 14, 10, 56, 13, tzinfo=tzutc())}, {'Name': 'parsing-xml-file', 'CreationDate': datetime.datetime(2024, 11, 13, 13, 3, 46, tzinfo=tzutc())}, {'Name': 'restaurant2024', 'CreationDate': datetime.datetime(2024, 8, 12, 11, 29, 26, tzinfo=tzutc())}, {'Name': 's3demobucketpy', 'CreationDate': datetime.datetime(2024, 12, 10, 10, 25, 1, tzinfo=tzutc())}, {'Name': 'sarangmybucket', 'CreationDate': datetime.datetime(2024, 11, 20, 22, 28, 15, tzinfo=tzutc())}, {'Name': 'sdggdggssdgag', 'CreationDate': datetime.datetime(2024, 11, 13, 18, 49, 25, tzinfo=tzutc())}, {'Name': 'ss-practice-matillion-revision', 'CreationDate': datetime.datetime(2024, 11, 20, 22, 37, 55, tzinfo=tzutc())}, {'Name': 'ssbucketdemo', 'CreationDate': datetime.datetime(2024, 10, 5, 6, 57, 28, tzinfo=tzutc())}, {'Name': 'test-glue-new', 'CreationDate': datetime.datetime(2024, 12, 21, 15, 35, 37, tzinfo=tzutc())}, {'Name': 'videosummarizer', 'CreationDate': datetime.datetime(2024, 3, 11, 19, 23, 6, tzinfo=tzutc())}], 'Owner': {'DisplayName': 'info', 'ID': '10954629e3dfaabc0680ca7e878fc30c5affd71bd871a8f63bdb28f675a874d9'}}

In [26]:
```python
# ACCESSING SPECIFIC BUCKET INFO

# Specify the name of your S3 bucket
bucket_name = 'czec-banking'
# List all objects in the specific S3 bucket
response = s3.list_objects_v2(Bucket=bucket_name)
# Print object keys (file names)
if 'Contents' in response:
    for obj in response['Contents']:
        print(f"Object Key: {obj['Key']}")
else:
    print("No objects found in the bucket.")
```

```
Object Key: ACCOUNT/
Object Key: ACCOUNT/account_new.csv
Object Key: CARD/
Object Key: CARD/card.csv
Object Key: CLIENT/
Object Key: CLIENT/client.csv
Object Key: DISPOSITION/
Object Key: DISPOSITION/disp.csv
Object Key: DISTRICT/
Object Key: DISTRICT/district.csv
Object Key: DOWNLOAD_CSV/
Object Key: DOWNLOAD_CSV/customer_data.csv_0_0_0.csv
Object Key: DOWNLOAD_CSV/sales_region_data.csv_0_0_0.csv
Object Key: INGESTION/
Object Key: INGESTION/ingested_data.csv
Object Key: LOAN/
Object Key: LOAN/loan.csv
Object Key: ORC_PARSER/
Object Key: ORC_PARSER/userdata1_orc
Object Key: ORC_PARSER/userdata2_orc
Object Key: ORC_PARSER/userdata3_orc
Object Key: ORC_PARSER/userdata4_orc
Object Key: ORC_PARSER/userdata5_orc
Object Key: ORDER_LIST/
Object Key: ORDER_LIST/order.csv
Object Key: TRANSACTIONS/
Object Key: TRANSACTIONS/trnx_16.csv
Object Key: TRANSACTIONS/trnx_17.csv
Object Key: TRANSACTIONS/trnx_18.csv
Object Key: TRANSACTIONS/trnx_19_NEW.csv
Object Key: TRANSACTIONS/trnx_20_NEW.csv
Object Key: TRANSACTIONS/trnx_21_NEW.csv
```

In [21]:
```
tables = ['sales_region_data', 'customer_data']  # Add all your table names here
```

In [25]:
```
# Export data to S3
for table in tables:
    file_name = f"{table}.csv"
    export_query = f"""
    COPY INTO @my_external_stage/{file_name}
    FROM (SELECT * FROM {table})
    OVERWRITE=TRUE;
    """
    conn.cursor().execute(export_query)
    print(f"Exported {table} data to S3 as {file_name}")

print("All tables exported successfully!")
```

```
Exported sales_region_data data to S3 as sales_region_data.csv
Exported customer_data data to S3 as customer_data.csv
All tables exported successfully!
```