CHAPTER 5 Data Ingestion : Loading Data

Sunday, 21 September 2025 00:08

<u>CONFIGURING AMAZON REDSHIFT WAREHOUSE AS A DESTINATION</u>

Configuring amazon redshift for your data warehouse, integration with S3 for loading data after it has been extracted.

<u>IAM Role Creation for Redshift</u>

- Go to IAM -> Roles -> Create new role
- Services ->Redshift (customizable)
- Attach Permission Policies -> AmazonS3readfullaccess
- Name -> RedshiftLoadRole
- Copy ARN (Amazon Resource Number) -> arn:aws:iam::448658737116:role/RedshiftLoadRole

- Create clusters in redshift

  [aws_creds] database = my_warehouse username = pipeline_user password = weifj4tji4j host =
  my_example.4754875843.useast-1.redshift.amazonaws.com
  port = 5439 iam_role = RedshiftLoadRole

<u>Loading Data into Redshift Warehouse from S3 Storage</u>

1. <u>Via SQL Queries in Redshift Query Editor</u>

The most efficient way to load data into redshift table is to use copy command with either SQL or python scripts using boto3. To query the cluster data easiest way to use is redshift query editor.

<u>COPY Command Syntax</u>

```
COPY table_name
[ column_list ]
FROM source_file
authorization
[ [ FORMAT ] [ AS ] data_format ]
[ parameter [ argument ] [, .. ] ]
```

<u>Execution</u>

```
COPY my_schema.my_table
FROM 's3://bucket-name/file.csv’
iam_role 'arn:aws:iam::222:role/RedshiftLoadRole’;
```

When this sql statement is executed the contents of files.csv are appended to a table called my_table in the my_schema schema of your redshift cluster.

Note : To specify the order in which you want to insert columns, you can do so by executing this statements with columns order

```
COPY my_schema.my_table (column_1, column_2, ....)
FROM 's3://bucket-name/file.csv'
iam_role 'arn:aws:iam::222:role/RedshiftLoadRole';
```

2. <u>Via Python Scripts in VS code using Boto3</u>

- Intsall psycopg2 (To interact with Redshift cluster from vscode using python)

- Create a file called copy_to_redshift.py

```
import boto3
import configparser
import psycopg2

# connect to redshift cluster
parser = configparser.ConfigParser()
parser.read("pipeline.conf")
dbname = parser.get("aws_creds", "database")
user = parser.get("aws_creds", "username")
password = parser.get("aws_creds", "password")
host = parser.get("aws_creds", "host")
port = parser.get("aws_creds", "port")

# connect to the redshift cluster
rs_conn = psycopg2.connect(
 "dbname=" + dbname
 + " user=" + user
 + " password=" + password
 + " host=" + host
 + " port=" + port)

# load the account_id and iam_role from the
# conf files
parser = configparser.ConfigParser()
parser.read("pipeline.conf")
account_id = parser.get("aws_boto_credentials",
 "account_id")
iam_role = parser.get("aws_creds", "iam_role")
bucket_name = parser.get("aws_boto_credentials",
 "bucket_name")

# run the COPY command to load the file into Red shift
file_path = ("s3://"+ bucket_name
 + "/order_extract.csv")
role_string = ("arn:aws:iam::"
 + account_id
 + ":role/" + iam_role)
sql = "COPY public.Orders"
sql = sql + " from %s "
sql = sql + " iam_role %s;"

# create a cursor object and execute the COPY
cur = rs_conn.cursor()
cur.execute(sql,(file_path, role_string))

# close the cursor and commit the transaction
cur.close()
rs_conn.commit()

# close the connection
rs_conn.close()
CREATE TABLE public.Orders (
 OrderId int,
 OrderStatus varchar(30),
 LastUpdated timestamp
);
```

- Execute (env) $ python copy_to_redshift.py

If the data was extracted in full, then you have one small addition to make to the loading script. Truncate the destination table in Redshift (using TRUNCATE) before you run the COPY operation

import boto3 import configparser import psycopg2 parser = configparser.ConfigParser() parser.read("pipeline.conf") dbname = parser.get("aws_creds", "database") user = parser.get("aws_creds", "username") password = parser.get("aws_creds", "password") host = parser.get("aws_creds", "host") port = parser.get("aws_creds", "port") # connect to the redshift cluster rs_conn = psycopg2.connect("dbname=" + dbname + " user=" + user + " password=" + password + " host=" + host + " port=" + port)