**1)Things to do in AWS**

**//Create a Bucket**

>> Create a Bucket with general configuration

>> Object ownership set to ACL’s Disabled

>> Block public access

>> Bucket versioning Disabled

>> Encryption key type >Amazon S3-managed keys (SSE-S3)

>> Bucket key enable

**//Create a folder in the object**

>> Folder name

>> Amazon S3-managed keys >SSE-S3

**//Creating a role using IAM**

>> Rloes

>> Create role

>> trusted entity type AWS Account

>> AWS Account> this account

>> options > Required external ID>00000

>> AmazonS3 Full access

>> role details

>> Create Role (snowconnect)

**2) Reading the parquet file using python and cleaning data**

>> import pyarrow.parquet as pq

trips = pq.read\_table('userdata1.parquet')

trips = trips.to\_pandas()

>> Cleaning the data using numpy

>> trips=trips.replace('',np.nan) // replacing null values with nan

>> trips['id']=trips['id'].replace(np.nan, 0) //as id is int replacing nan values with 0

>> trips['salary']=trips['salary'].replace(np.nan, 0) //as id is int replacing nan values with 0

**// Saving the dataframe as parquet file**

>> trips.to\_parquet('employee1.parquet')

**3) Loading into S3 bucket**

>> Load the saved parquet file into the folder

**4) In snow flake**

**//create data base**

Create database nkbase;

**//using the database**

use database nkbase;

**//creating the schema**

create schema parquetschema;

**//using the schema**

Use schema parquetschema

**//granting permission**

GRANT USAGE ON Database nkbase TO ROLE SYSADMIN;

**//creating integration b/w aws and anow flake**

CREATE or replace STORAGE INTEGRATION parquet\_integration

TYPE = EXTERNAL\_STAGE

STORAGE\_PROVIDER = S3

ENABLED = TRUE

STORAGE\_AWS\_ROLE\_ARN ='arn:aws:iam::982637298554:role/parquetconnect'

STORAGE\_ALLOWED\_LOCATIONS = ('s3://nithyaprivateparquet/employee1.parquet');

**//listing the integration**

desc integration parquet\_integration;

**//creating file format object**

CREATE or replace FILE FORMAT myparquetformat

TYPE = 'PARQUET'

SNAPPY\_COMPRESSION = TRUE

BINARY\_AS\_TEXT = TRUE

TRIM\_SPACE = TRUE

**// creating a table**

create or replace table userdata(

id int,

first\_name varchar,

last\_name varchar,

email varchar,

gender varchar,

ip\_address varchar,

cc varchar,

country varchar,

birthdate varchar,

salary float,

title varchar,

comments varchar);

**5) Then in AWS we need to update the policy**

>>Roles

>> Our Created role (snow connect)

>> Trust Relationships

>> Edit trust policy

>> Replace arn and external id with our created roles arn and externl id

>> update policy

**6) In snowflake**

**//creating internal stage**

create or replace stage parquetstage

url = 's3://nithyaprivateparquet/employee1.parquet'

storage\_integration = parquet\_integration

file\_format = myparquetformat

**//viewing the stage**

list @parquetstage;

**//coping into table**

COPY INTO userdata

FROM @parquetstage

match\_by\_column\_name = case\_insensitive

ON\_ERROR = 'CONTINUE';

select \* from userdata;