**Integrating AWS & SnowFlake and Creating a Snow pipe**

**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

>> Load the csv file into the folder

**//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)Things to do in snowflake**

**// creating a database**

create database nkbase;

**// using the database**

use database nkbase;

**// creating a schema**

create schema nkschema;

**// Giving permission to sysadmin**

GRANT USAGE ON Database nkbase TO ROLE SYSADMIN;

**//This query is used to integrate the object storage in Aws to snowflake by using IAM policy**

CREATE or replace STORAGE INTEGRATION flights\_integration

TYPE = EXTERNAL\_STAGE

STORAGE\_PROVIDER = S3

ENABLED = TRUE

STORAGE\_AWS\_ROLE\_ARN ='arn:aws:iam::975362200552:role/snowconnect'

STORAGE\_ALLOWED\_LOCATIONS = ('s3://nkasam/flightsdump/');

**// Gives the description of the file**

Desc integration flights\_integration;

**//This is a file format object**

create or replace file format newformat

type = CSV

field\_delimiter = ','

skip\_header = 1 -- skip the first row

--field\_optionally\_enclosed\_by = '"'

-- for multiple lines of data separated by comma engrossed inside''

empty\_field\_as\_null = True;

**//Creating table**

create or replace table flightsdata(

slno int,

airline varchar,

flight varchar,

source\_city varchar,

dept\_time varchar,

stops varchar,

arrival\_time varchar,

dest\_city varchar,

class varchar,

duration float,

days\_left varchar,

price int

);

**3)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

**4) In snowflake**

**//creating an internal stage**

create or replace stage nkstage

url = 's3://nkasam/flightsdump/'

storage\_integration = flights\_integration

file\_format = newformat

**//view the csv file present in stage**

list @nkstage

**//copying data from stage to raw table**

OPY INTO flightsdata

FROM @nkstage

ON\_ERROR = 'skip\_file';

**//View the data in the table**

select \* from flightsdata;

**// Create a snow pipe**

create or replace pipe flightpipe

auto\_ingest = true

as

copy into flightsdata

from @nkstage;

**// Describing snow pipe**

desc pipe flightpipe;

>> copy the notification channel

**5) In AWS**

>> go to created bucket

>>Properties

>>create an event notification

>> enter event name

>>Object creation (check all object creation)

>> Destination > SQS Queue

>> Specify SQS Queue > Enter SQS > paste the copied notification channel

>> Upload another data set into the S3 bucket folder where the old dataset is present

**6) In Snowflake**

>> Select count (\*) from flightsdata;

\* This appends the data present in the second dataset to the first dataset.