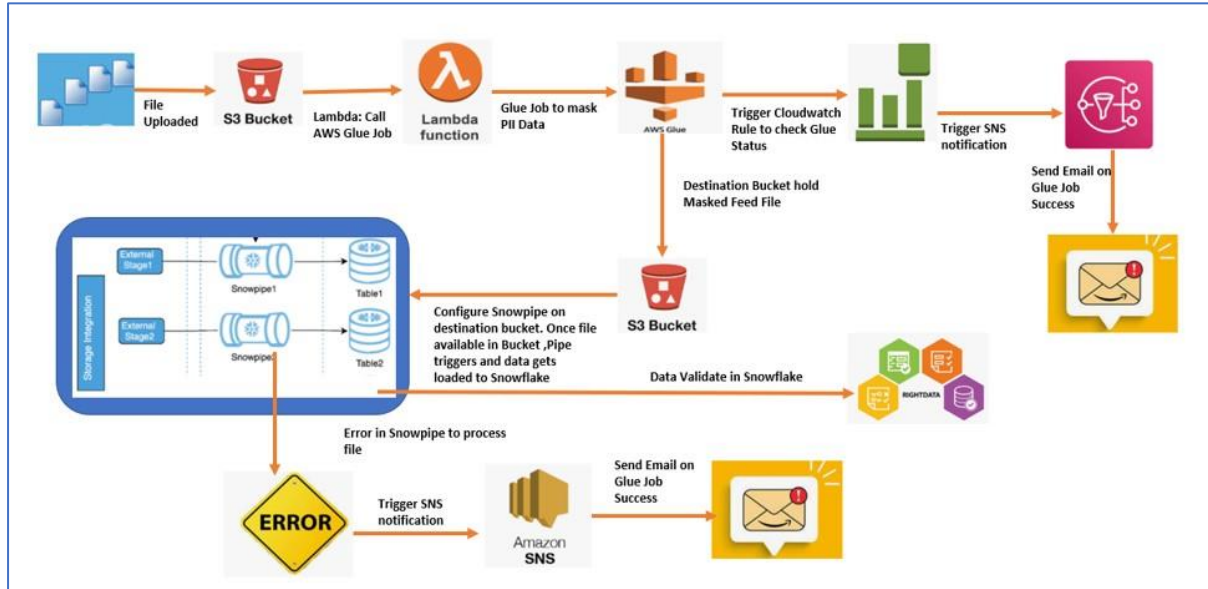


# Snowflake: Data pipeline with Glue and Snowpipe

During this post we will discuss about below end-to-end pipeline.



## Glue and Pipe Data Pipeline

As we can see in above Data pipeline, we have used several AWS and Snowflake components at one place.

[AWS Glue](#) is a Serverless Extract, Transform, and Load (ETL) service combines the speed and power of Apache Spark. Moreover, we have Used Lambda function to automate and Call the [AWS Glue Service](#). However, We have leveraged the Cloud watch events to define a Rule based on the status of Glue job. Finally, AWS SNS service is used to notify the stakeholders about Glue job.

Finally, Configure Snowpipe, consumes the file uploaded by Glue to the Bucket and ingest to the Snowflake. Implement snowflake Error notification which will send email in case of pipe failure.

### Steps performed in the Pipeline:

- Firstly, Customer Feed file uploaded to the Source bucket.
- Once the File uploaded, Lambda gets trigger and call the AWS Glue job.
- Once the AWS Glue job runs it masks the data inside the Feed file.
- Therefore, Glue Job place this new file to the destination bucket.
- Moreover, Configure the Cloud watch Event Rules to check the status of Glue Job.
- Therefore, based on the Glue Job Status, SNS will be triggered to send an email to respective stakeholders.

- Snowpipe configure on the destination bucket and trigger once file uploads to Bucket.
- Finally, Data gets ingest to the Snowflake.
- Above all, Configure the Snowpipe Error Notification to check the status of Snowpipe Job.
- In case of any error to the Snowpipe, SNS will be triggered to send an email to respective stakeholders.

### Technical implementation (Code) of Data Pipeline:

**Source bucket:** gluemaskingpiisrcfeed

**Lambda function:** Glue\_Masking\_Lambda

**GlueJob:** Job\_Masking\_PII\_Data

**Destination bucket:** gluedatamasking

**CloudWatch Rules:**Masking\_Glue\_job\_notify

**SNS:**EC2\_LOGS\_Email\_Send

**SnowPipe:**glue\_snow\_load

**Snowpipe Error notification:**Snowpipe\_Error\_Notify

**SNS:**snowpipe\_alert

-

### Lambda Function:

```
import json

import boto3

def lambda_handler(event, context):

    s3_client = boto3.client("glue");

    s3_client.start_job_run(JobName="Job_Masking_PII_Data")

    return {

        'statusCode': 200,

        'body': json.dumps('Hello from Lambda!')

    }
```

### Glue Script:

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
```

```

def mask(dynamicRecord):
dynamicRecord['phone'] = '*****'
dynamicRecord['ssn'] = '*****'
return dynamicRecord

args = getResolvedOptions(sys.argv, ['JOB_NAME'])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
import boto3
client = boto3.client('s3')
BUCK_NAME = "gluedatamasking"
PREFIX = "run-"
response = client.list_objects(
Bucket=BUCK_NAME,
Prefix=PREFIX,
)
name = response["Contents"][0]["Key"]
client.delete_object(Bucket=BUCK_NAME, Key=name)
job.commit();
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "pii",
table_name = "pii_customer_invoice_csv", transformation_ctx = "datasource0")
masked_dynamicframe = Map.apply(frame=datasource0, f=mask)
bucket_name = "gluedatamasking"
datasink4 = glueContext.write_dynamic_frame.from_options(frame =
masked_dynamicframe, connection_type = "s3", connection_options = {"path":
f"s3://{bucket_name}/"}, format = "csv", transformation_ctx = "datasink4")

```

## Snowflake:

```

create or replace table GLUE_SNOWPIPE_TBL (
SSN VARCHAR(20),
CRID VARCHAR(4),
INV_AMT number(10) null,
CUST_BAL number(20) null,
CUST_NUM varchar(50) null,
CUST_STAT varchar(255) null,
INV_NO varchar(10) null,
Email VARCHAR(50),
phone varchar(20)
);

```

```

CREATE NOTIFICATION INTEGRATION Snowpipe_Error_Notify
ENABLED = true
TYPE = QUEUE
NOTIFICATION_PROVIDER = AWS_SNS
DIRECTION = OUTBOUND
AWS_SNS_TOPIC_ARN = 'arn:aws:sns:us-east-1:913267004595:snowpipe_alert'
AWS_SNS_ROLE_ARN = 'arn:aws:iam::913267004595:role/snowpipe_SNS_Error_Role'

create or replace stage demo_db.public.glue_snowpipe_stage
URL = 's3://gluedatamasking/'
STORAGE_INTEGRATION = s3_int
file_format = demo_db.public.csv_format;

CREATE or replace PIPE demo_db.public.glue_snow_load auto_ingest=true
ERROR_INTEGRATION = Snowpipe_Error_Notify
AS copy into GLUE_SNOWPIPE_TBL from @demo_db.public.glue_snowpipe_stage/

```

## Error Notify

### Details

snowpipe\_alert

snowpipe\_alert

arn:aws:sns:us-east-1:913267004595:snowpipe\_alert

913267004595

Standard

### Access policy

**Data protection policy**

Delivery retry policy (HTTP/S)

### Delivery status logging

## Encryption

## Tags

## Edit


Delete

Request confirmation

### Confirm subscription

Create subscription

Q Search

< 1 > 

ID



Endpoint

▽

Status



### Protocol

▲

f03c622e-e392-46d6-85d8-540eff12cb56

sachin.mittal04@gmail.com

Confirmed

EMAIL-JSON

```
27 select * from GLUE_SNOWPIPE_TBL
28
```

Results Data Preview

← Opri

✓ Query\_ID SQL 1.4s  104 rows

Filter result...



Copy

Column

Row	SSN	CRID	INV_AMT	CUST_BAL	CUST_NUM	CUST_STAT	INV_NO	EMAIL	PHONE
1	"*****"	2235	20000	300000	C-100	Active	I-100	C-100@support.com	"*****"
2	"*****"	2234	30000	200000	C-100	Active	I-101	CC-100@support...	"*****"
3	"*****"	2234	50000	500000	C-100	Active	I-102	C-100@support.com	"*****"
4	"*****"	2234	50000	500000	C-100	Active	I-103	C-100@support.com	"*****"
5	"*****"	2234	50000	500000	C-100	Active	I-104	C-100@support.com	"*****"
6	"*****"	2234	300000	500000	C-100	Active	I-105	C-100@support.com	"*****"
7	"*****"	2234	50000	100000	C-101	Active	I-113	C-101@support.com	"*****"
8	"*****"	2234	50000	100000	C-101	Active	I-114	C-101@support.com	"*****"
9	"*****"	2234	40000	50000	C-102	Active	I-115	C-102@support.com	"*****"
10	"*****"	2234	10000	50000	C-102	Active	I-116	C-102@support.com	"*****"
11	"*****"	2234	30000	80000	C-103	Active	I-117	C-103@support.co...	"*****"
12	"*****"	2234	50000	80000	C-103	Active	I-118	C-103@support.co...	"*****"

## Data in SF