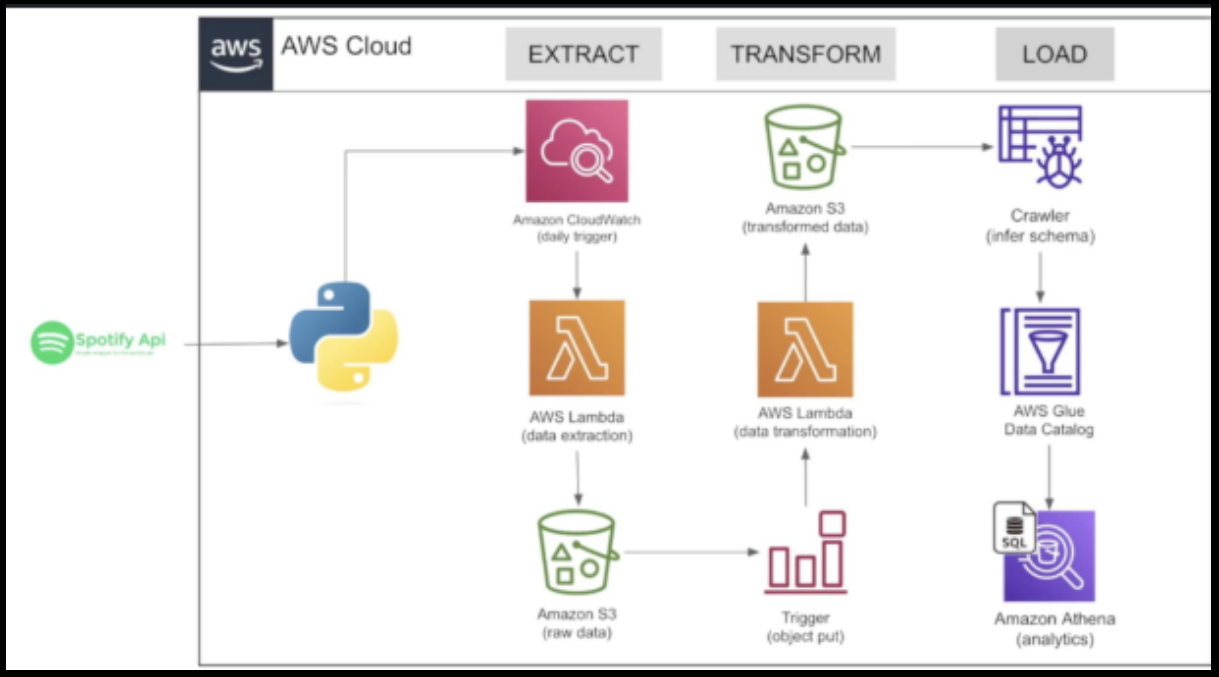
1.SPOTIFY DATA EXTRACTION AND LOADING

Business probelm: User wants to explore weekly trending songs and improve the app based on it.

-collect the data on a weekly basis so that after one or two years he can find different insights ;so we need to build an ETL pipeline for it.

Initially we are gonna build this pipeline:



EXTRACTION:
Spotify api- data extraction.
AWS LAMBDA: Deploy code in aws lambda
AWS CLOUDWATCH: we can schedule ,to run lambda on particular interval we provide
First we collect data from spotify API and store it in s3(raw data)
TRANSFORM:
+ Add trigger
AWS LAMBDA(data transformation)
= s3(transformed data)
Whole process is automated
Then
LOAD:
GLUE CRAWLER: Go through each column it understand the data ,diffrent column names,data type and it will build the GLUE CATALOG(all details)
ATHENA : Directly run sql queries.

SPOTIFY API : Music artists,albums, and tracks

AMAZON S3:
-simple storage service
-storage service that can store and retrieve any amount of data from anywhere on the web.
-used to store and distribute large media files,data backups,and static website files.

COMPUTE:(AWS LAMBDA):
-Serverless computing service.
- To run code response to events like changes in s2,dynamodb,or other aws services.

LOGS/TRIGGER:
-Collect and monitor log files,alarms

PROJECT ::

1.Open https://developer.spotify.com/dashboard
Create app
App name:
Client id : b6e073250b1044eca162cee8c1e53ec9
Client secret :8c2e08d41bed40678503fd92a7bd10af

```
client_credentials_manager = SpotifyClientCredentials(client_id =
"b6e073250b1044eca162cee8c1e53ec9",client_secret="8c2e08d41bed40678503f
d92a7bd10af")
```

This is the authentication

```
sp = spotipy.Spotify(client_credentials_manager =
client_credentials_manager)
```

This is authorization.

Copy pasted the URL link

From URL link extracted the id of it

From the data we are extracting the different lists from that we are converting it into dictionary for json format

Next we are creating a list
List with pandas we create a dataframe
Changed date time and maintained correct data types

CREATE S3 BUCKET- WE WILL STORE THE DATA
-create raw and transformed folders

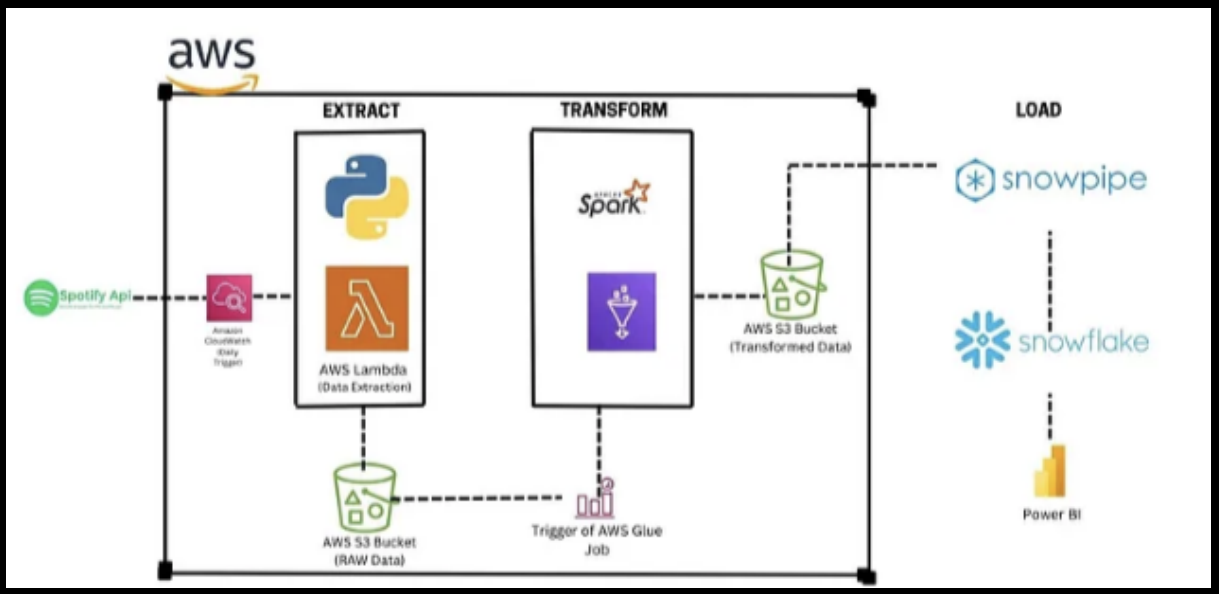
CREATE LAMBDA TRANSFORMATIONS
-CREATE EXTRACT FUCNTION
Lambda - add the code
1.you will get one problem like pandas not installed so for aws lambda you should do following process ;
Lambda→ layers→ add the xip file and create a layer and then add it in the code choose custom layer

Import boto3

Entyire json file to s3

Mow we are gonna automate this by event bridge

In python for data engineering course transformation in labda itself.Bu**t well write the transformations in spark.**



AWS GLUE:

Visual ETL
NOTEBOOKS
JOB RUN MONITORING
CATALOG TABLES
DATA CONNECTIONS
WORKFLOWS

Now main project::

1. **Using Apache spark we will write the transformation and then we will deploy it into AWS GLUE**
2. **Create IAM ROLES: GIVE S3 FULL ACCESS GLUE ACCESS NOTEBOOK ACCESS as of now**
3. **Attach the IAM role in the glue**
4. **Now the notebook is created**

In notebook first you will get IAM: pass role so you add the IAM full access as well to the roles.
-Imported packages.

1. **We need to read the data from s3 and copy the uri path and earlier in python we had to use boto3 and then read it,but in glue it natively reads the data from s3 .s3_path ="s3://spotify-etl-complete-project/raw_data/to_processed/"**
2. **source_dyf = glueContext.create_dynamic_frame_from_options(same like spark.get create stuff in python) but using glue concept**
3. **We need to convert this json into a data frame so that we can easily process it toDF().**

—- We read data from s3 bucket into here—---

4. **Using col,explore functions.**
5. **We already got the df now extract each column from it as we done it in python here by using col**
6. **We have to extract album artist songs using spark the only diff is in python we use loops and all that complicated shit but here we just use the explode function**
7. **Converted into df,functions**

— processing is done—-

8. **Now write this file into s3**
9. **Spark – into dynamic frame**
10. **Now we are creating a function to write to s3**
    Then these files get transformed into s3

    —------ upto transformation glue it is completed—---

NOW SNOWFLAKE SIDE::

1. **Create a database**
2. **To give access/ to connect to s3 to the snowflake go to IAM ROLE (snowflake side):**
   CREATE DATABASE spotify_db;
3.
4. **CREATE OR REPLACE storage integration s3_init**
5. **TYPE = EXTERNAL_STAGE**
6. **STORAGE_PROVIDER = S3**

7. ENABLED = TRUE
8. STORAGE_AWS_ROLE_ARN = 'arn:aws:iam::241533125232:role/spotify-spark-snowflake-role'
9. STORAGE_ALLOWED_LOCATIONS = ('s3://spotify-etl-complete-project')
10. COMMENT = 'Creating connection to S3'
11.
12. DESC integration s3_init; from this description you will be able to get

STORAGE_AWS_IAM_USER_ARN

STORAGE_AWS_EXTERNAL_ID

Copy these both and update in ther iam trust relationships where we connect or give access to connect this s3 and snowflake we are giving access
— connection is created for the snowflake and s3 —---

13. FILE FORMAT
14. Stage
    - create the actual connection to using storage integration to s3
15. Three tables album artist songs artist are created
16. Using copy into command to fill the data in the tables

—-- snowpipe creation (for automation)-----

```
CREATE OR REPLACE pipe pipe.tbl_album_pipe
auto_ingest = TRUE
AS
COPY INTO tbl_album
FROM @spotify_stage/album/;
```

So whenever in transformed data is uploaded then in this snowflake table is updated with this copy command

1. Use DESC and copy the notification channel sqs(simple queue service notification channel - any event /update is done in s3 is notified with this sqs into snowflake)
2. In aws side go to the s3 bucket -properties– event notification–
3. To check if any errors in snowflake pipe SELECT SYSTEM$PIPE_STATUS('pipe.tbl_songs_pipe');
   Same for album,artists

Triggered the aws glue as well now

You can move the processed files to proceesdd_processed by adding the code in etl script
list

In lambda instead of event clous watch we directly added the glue job in lambda itself when we run it it automatically starts running in the glue we can see it there as well now we add cloud watch trigger at the data extraction i.e lambda function then whole pipe is automated.