E-Commerce Analysis

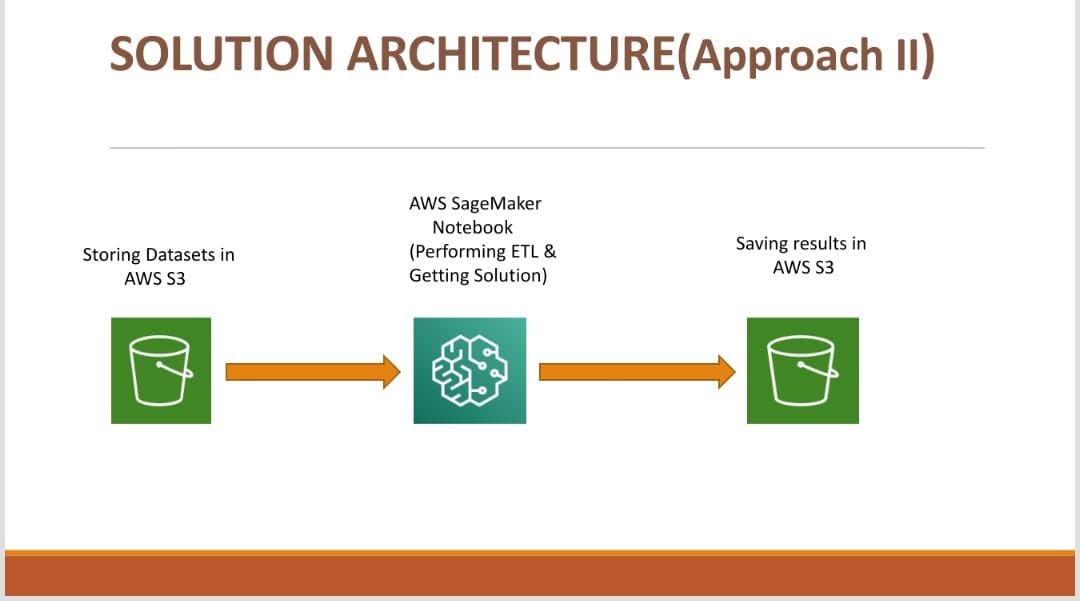
Objective- Analyzing E-Commerce dataset to draw insights about sales pattern across different geographic regions and locations. Expanding Customer reach by improving the sales of the business by identifying most valued regions and products.

Software / Tools used –

1. Python Notebook – A python notebook is an open source web application that you can use to create and share live code, equations, visualizations and test. Python notebooks are provided by different organizations under different name – Jupyter Notebook, DataBricks(Azure Cloud), Sagemaker(AWS). But all the notebooks have same properties and same code can be run anywhere. It is saved as a .ipynb file.
2. Storage Data Lake – Data lake is a centralized repository that allows you to store structured and unstructured data at any scale. In AWS we use S3 storage for this purpose as we can connect multiple sources with the same. (our hard drive can also act as same if cloud not avaliable)
3. ETL Software – ETL stands for extract, Transform and Load meaning we form metadata or structures of the dataset so that we can run queries on the same using sql. In AWS we use Glue, it creates metadata using crawlers and defines a schema for a dataset file and then we can run queries over the same using a query engine. (In local machine we can use SSMS Import function which imports the file to a table by automatically identifying the schema and creating a table for us to run queries on)
4. Query Engine – A query engine is a piece of software that sits on top of database or server and executes queries against data in that database or server. It makes it easy to get the desired result using simple SQL queries. In AWS we have Athena and Redshift for the same purpose. (Locally we can use SSMS or MySql Server to carry out the same tasks)
5. AWS Glue Studio (Optional)- we can also tailor a job to run a query on a database and store the results as desired format in a format or location of our choosing.

Approach 1-

Flow diagram:



Procedure Followed:

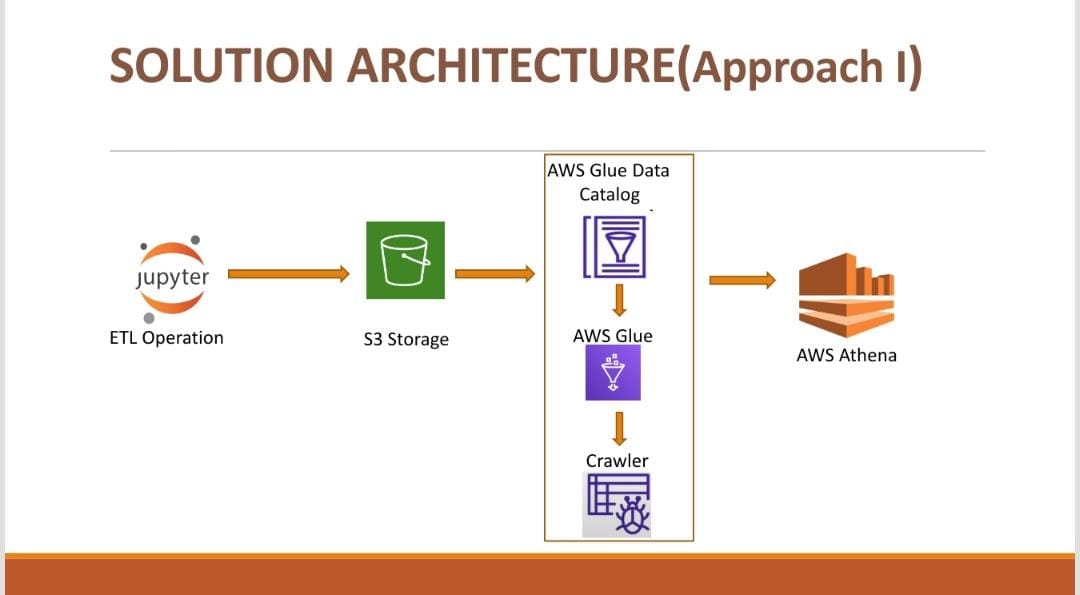
1. Load the data in S3 so that it can be speedily accessed through hdfs.
2. Save the stored path to the documents to load the files later
3. (Specific for AWS, alternatively we can open any python notebook) create an endpoint and give it a suitable IAM role to access objects in S3 bucket.
4. Create a notebook using this endpoint and launch the same and code to perform data clensing and find the solutions for the problems given

In the Python Notebook-

1. Check the requirements of packages to be used and install if any not present using pip
2. Now import the libraries which will be used in our ETL as well as analysis and visualisation.
3. Start a spark session to use its capabilities to form a data frame.
4. Declare the variables which we will require frequently to be changed.
5. Now one by one import all the datasets using spark.read.csv() functionality and create a data frame, clean the data frame for the analysis of the business cases.
6. Apply different filter and joins to answer the business solutions.
7. Use interactive input to answer the cases which require user input from the user.
8. We can also use visualisations through matplotlib and pygal to form charts and graphs that make the display of data easier.
9. You can see the solution to the problems with attached file here <Pyspark_Approach_1.html> (Pyspark\_Approach\_1.html)

Approach 2-

Flow diagram:



Procedure Followed:

1. Data is loaded into a python notebook and performed cleaning upon.
2. The clean data is saved in the Data lake Again (S3 in our case).
3. The data has to be read and be made ready to run query upon
4. AWS Glue Crawlers crawl through the data and make meta data tables.
5. Metadata tables just define the structure of the file and data is read according to this format.
6. The file is stored in the hadoop only but is now readable according to the metadata structure.
7. Now we have table structure so it is easier to run sql queries using athena.

Data cleaning using Python notebook:-

1. We load the required libraries into the the notebook and create a spark object
2. Now the spark object is used to create a pyspark dataframe object for of the datsets
3. The data set is now cleaned using different operations for each.
4. The cleaned dataframe is now saved as csv again in the storage for further operations.
5. Refer file <ETL_Approach_2.html> (ETL\_Approach\_2.html)

AWS Glue Crawlers for making metadata:

1. AWS Glue helps us to make the meta data tables and the table catalogue, ie. It creates a table like structure for the data file so that we can query data using simple sql like queries.
2. First we have to setup the crawlers which point to the file for which we have to create meta data
3. We provide appropriate permissions to the crawlers to access the data files in data lakes using IAM roles and appropriate policies.
4. The crawlers run serverless and create a metadata table with the structure required and is then saved in a database.
5. We can see the data that is produced by the table by querying the table using athena.
6. But in reality the table doesn’t contain the information but only the structure and the information is returned by going over the file actually in the HDFS or the local storage
7. Refer <Queries_Approach_2.docx> (Queries\_Approach\_2.docx)

Querying the data(Solutions) by Athena:

1. Open Athena Query engine.
2. It shows the databases present in our account.
3. Select the database we have our tables saved in.
4. Now query in sql to get the answers to each business problem.
5. Use joins, unions and views to get our desired result.
6. The output can be saved to a csv file in a destination folder of our selection.
7. Refer <Queries_Approach_2.docx> (Queries\_Approach\_2.docx)

NFRs:

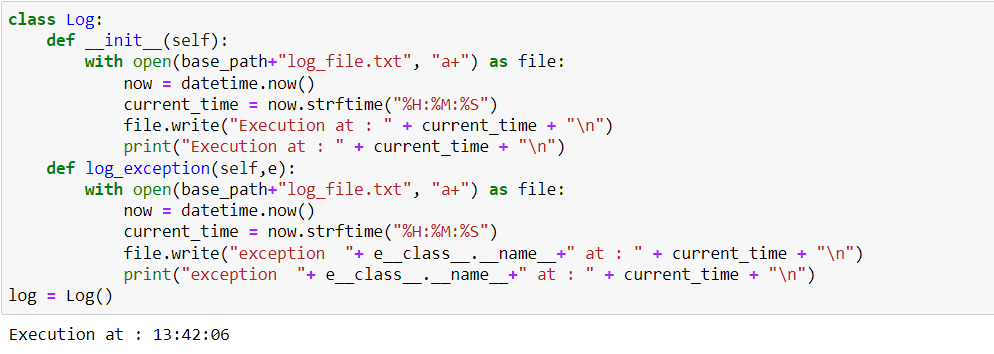
1. Logging Framework
2. Exception Handling



We log every exception which helps us better find where a certain code run failed and why exactly that problem occurred.

Whenever an exception arises it doesn’t stop the program execution whereas it is properly caught and logged for better use.

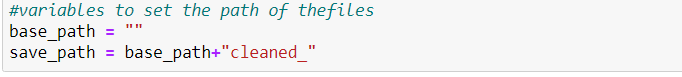
1. OOPS Implemantation



We created a class for logs which handles all actions for the logging at one place.

An object of the same class can be used for calling of different type of functions related to it

1. Parameterised Implementation



Some variables are used everywhere and have to be changed frequently so instead of changing the code everywhere we can just change the value of the given varaiable and this will make the changes in your entire code.

**Refer to** [**presentation\_spark\_301.pptx**](presentation_spark_301.pptx) **(presentation\_spark\_301.pptx) for a broader view**