**Lab Exercise- Using AWS Glue Notebooks with Amazon S3**

Here's a step-by-step guide to help you with this exercise:

**Objective:**

Learn how to use AWS Glue Notebooks to perform ETL operations on data stored in Amazon S3.

**Pre-requisites:**

* AWS account
* Basic knowledge of AWS Glue, Amazon S3, and Python

**Part 1: Set Up AWS Glue Environment**

**Log in to AWS Management Console:**

* Navigate to the AWS Glue service.

**Create an S3 Bucket:**

* Go to the S3 service and create a new bucket (e.g., my-glue-bucket).
* Upload some sample data files (e.g., CSV files) to this bucket.

**Create an IAM Role for AWS Glue:**

* Go to the IAM service and create a new role.
* Select the "Glue" service for the role type.
* Attach the necessary policies (e.g., AmazonS3FullAccess, AmazonGlueServiceRole).
* Note the ARN of the role.

**Part 2: Create an AWS Glue Notebook**

**Navigate to AWS Glue Studio:**

* Go to the AWS Glue service and select "AWS Glue Studio."

**Create a New Notebook:**

* Click on "Notebooks" in the sidebar.
* Click "Create notebook."
* Enter a notebook name (e.g., MyGlueNotebook).
* Select the IAM role you created earlier.
* Choose the notebook type (e.g., Python).

**Open the Notebook:**

After creating the notebook, open it to start writing code.

**Part 3: Write ETL Code in the AWS Glue Notebook**

Import Required Libraries:

In the notebook, import the necessary libraries:

import boto3

import pandas as pd

from pyspark.sql import SparkSession

**Initialize Spark Session:**

**Create a Spark session to process data:**

spark = SparkSession.builder.appName("AWS Glue Notebook").getOrCreate()

**Read Data from S3:**

Load data from your S3 bucket into a Spark DataFrame:

s3\_bucket = "my-glue-bucket"

file\_path = "s3://{}/your-data-file.csv".format(s3\_bucket)

df = spark.read.csv(file\_path, header=True, inferSchema=True)

**Perform Data Transformations:**

Example transformation: Filter rows and select specific columns:

transformed\_df = df.filter(df["some\_column"] > 100).select("column1", "column2")

**Write Data Back to S3:**

Save the transformed data to another location in your S3 bucket:

output\_path = "s3://{}/transformed-data/".format(s3\_bucket)

transformed\_df.write.mode("overwrite").csv(output\_path, header=True)

**Run the Notebook:**

* Execute the cells in the notebook to perform the ETL operations.

**Part 4: Verify the Results**

**Check S3 Bucket:**

* Go back to the S3 service and navigate to the output path you specified (transformed-data/).
* Verify that the transformed data files are saved correctly.