**Hands-On with AWS Analytics Services**

**Prerequisites**

1. An active AWS account with sufficient permissions to create resources.
2. Data for analysis. (You can use a sample dataset like the NYC Taxi Dataset available on AWS S3.)
3. Basic knowledge of SQL and data visualization concepts.

**Part 1: Querying Data with Amazon Athena**

**Objective: Use Amazon Athena to query data stored in S3 using SQL.**

1. **Upload Sample Data to S3**:
   * Go to the **S3** service and create a bucket (e.g., aws-analytics-demo-bucket).
   * Upload a sample CSV dataset (e.g., nyc\_taxi\_data.csv) to the bucket.
2. **Create an Athena Table**:
   * Open the **Amazon Athena** console.
   * Navigate to the **Query Editor**.
   * Set up an **S3 bucket** for query results (e.g., aws-analytics-demo-results).
   * Run a query to create a table:

sql

Copy code

CREATE EXTERNAL TABLE nyc\_taxi\_data (

vendor\_id STRING,

pickup\_datetime TIMESTAMP,

dropoff\_datetime TIMESTAMP,

passenger\_count INT,

trip\_distance DOUBLE,

fare\_amount DOUBLE

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 's3://aws-analytics-demo-bucket/';

1. **Query the Data**:
   * Run a simple query to analyze data:

sql

Copy code

SELECT vendor\_id, AVG(fare\_amount) AS avg\_fare

FROM nyc\_taxi\_data

GROUP BY vendor\_id;

1. **Visualize Results**:
   * Export the query results to CSV or integrate with QuickSight for visualization.

**Part 2: Data Visualization with Amazon QuickSight**

**Objective: Visualize the Athena query results.**

1. **Set Up Amazon QuickSight**:
   * Open the **Amazon QuickSight** console and sign up for the service.
   * Set up a connection to the Athena database.
2. **Create a Dataset**:
   * Import the Athena query results into QuickSight.
   * Use the **SPICE** engine for faster queries (optional).
3. **Build a Dashboard**:
   * Use the imported dataset to create visualizations:
     + Bar chart for vendor\_id vs. avg\_fare.
     + Line chart for trip distances over time.
   * Customize with filters and calculated fields.
4. **Share the Dashboard**:
   * Share the dashboard with team members or download the visualizations.

**Part 3: Data Transformation with AWS Glue**

**Objective: Use AWS Glue to clean and transform data.**

1. **Create a Glue Crawler**:
   * Go to the **AWS Glue** console.
   * Create a new crawler to scan the S3 bucket (aws-analytics-demo-bucket) and create a schema in the AWS Glue Data Catalog.
2. **Set Up a Job for Transformation**:
   * Create an ETL job in AWS Glue:
     + Use the data catalog as the source.
     + Transform data (e.g., drop unnecessary columns, clean null values).
   * Use PySpark for advanced transformations:

python

Copy code

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

args = getResolvedOptions(sys.argv, ['JOB\_NAME'])

sc = SparkContext()

glueContext = GlueContext(sc)

spark = glueContext.spark\_session

job = Job(glueContext)

job.init(args['JOB\_NAME'], args)

# Load data

datasource = glueContext.create\_dynamic\_frame.from\_catalog(

database="nyc\_taxi",

table\_name="nyc\_taxi\_data"

)

# Transformation

transformed = datasource.drop\_fields(['unnecessary\_column'])

# Write back to S3

glueContext.write\_dynamic\_frame.from\_options(

frame=transformed,

connection\_type="s3",

connection\_options={"path": "s3://aws-analytics-demo-transformed/"},

format="parquet"

)

job.commit()

**Part 4: Data Warehousing with Amazon Redshift**

**Objective: Load transformed data into Redshift for analysis.**

1. **Create a Redshift Cluster**:
   * Open the **Amazon Redshift** console.
   * Launch a new cluster and configure nodes.
   * Use the **Query Editor** or connect via a SQL client.
2. **Load Data into Redshift**:
   * Create a table in Redshift:

sql

Copy code

CREATE TABLE nyc\_taxi\_data (

vendor\_id VARCHAR(10),

pickup\_datetime TIMESTAMP,

dropoff\_datetime TIMESTAMP,

passenger\_count INT,

trip\_distance FLOAT,

fare\_amount FLOAT

);

* + Load data from S3 using the COPY command:

sql

Copy code

COPY nyc\_taxi\_data

FROM 's3://aws-analytics-demo-transformed/'

IAM\_ROLE 'arn:aws:iam::your-account-id:role/RedshiftS3AccessRole'

FORMAT AS PARQUET;

1. **Analyze the Data**:
   * Perform advanced SQL queries for analysis.

sql

Copy code

SELECT vendor\_id, COUNT(\*) AS total\_trips

FROM nyc\_taxi\_data

GROUP BY vendor\_id

ORDER BY total\_trips DESC;

1. **Integrate with BI Tools**:
   * Use tools like Tableau or QuickSight for further visualization.

**Summary**

This hands-on guide covers:

* Querying data with **Amazon Athena**.
* Visualizing results using **Amazon QuickSight**.
* Transforming data with **AWS Glue**.
* Warehousing data in **Amazon Redshift**.