Project Details

Superstore Data Analysis | End-to-End AWS Data Engineering Project

Dheeraj S Kulkarni

March 2025

Context

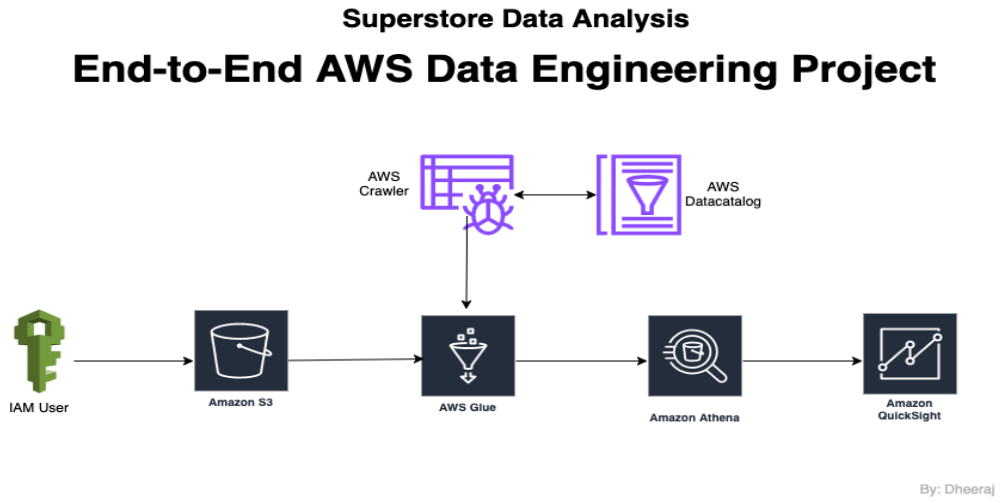
With growing demands and cut-throat competition in the market, a Superstore Giant is seeking our knowledge to understand what works best for them. They would like to understand which products, regions, categories and customer segments they should target or avoid.

Project Description

The “End-to-end AWS Data Engineering Project" provides a comprehensive walkthrough of building a data engineering project using Amazon Web Services (AWS). Throughout the project, the following AWS services are utilized:

1. **Identity and Access Management (IAM):** For managing user permissions and security.
2. **Simple Storage Service (S3):** To store and retrieve data files.
3. **AWS Glue:** For data cataloging and crawlers.
4. **Amazon QuickSight:** For data visualization and business intelligence.

Project Workflow



Resources

- **YT Source video:** Superstore Data Analysis | End to End AWS Data Engineering Proj...
- **Instructor:** Ankit Bansal
- **Dataset:** [Superstore Dataset]

IAM User

An IAM User represents an **individual person or application** with permanent credentials.

Key Features of IAM Users

- Has a username and password for AWS Console login.
- Can have access keys (Access Key ID & Secret Key) for API/CLI access.
- Permissions are controlled via IAM policies (e.g., Read/Write to S3).
- Used for long-term access to AWS resources.
- Cannot be assumed like roles; each user has dedicated credentials.

Amazon S3 (Simple Storage Service)

Amazon S3 (Simple Storage Service) is a **scalable, highly durable, and secure object storage service** that allows you to store and retrieve any amount of data from anywhere.

Querying Data in S3 with Athena

Amazon Athena allows you to run SQL queries on S3 data **without moving it**.

Steps to Query Data

1. Store **structured data** (CSV, JSON, Parquet) in S3.
2. Use **AWS Glue Crawler** to catalog the data.
3. Query using **Amazon Athena**.

AWS Glue

AWS Glue is a fully managed **serverless data integration service** from Amazon Web Services (AWS). It is designed to **extract, transform, and load (ETL)** data from various sources and prepare it for analytics, machine learning, and other data-driven applications.

Key Features of AWS Glue

1. **Serverless ETL** – No need to manage infrastructure; AWS Glue automatically provisions resources.
2. **Data Catalog** – Centralized metadata repository for structured and semi-structured data.
3. **Job Scheduling & Orchestration** – Automate and schedule ETL jobs.
4. **Support for Multiple Data Sources** – Works with Amazon S3, RDS, Redshift, DynamoDB, and more.
5. **Python & Scala Support** – Uses **Apache Spark** under the hood.
6. **Schema Discovery** – Crawlers automatically infer schema from raw data.

How AWS Glue Works

1. **Crawl Data Sources** – AWS Glue **Crawlers** scan and catalog metadata from various sources.
2. **Transform Data with ETL Jobs** – Use **PySpark** or **Scala** to clean, filter, and transform data.
3. **Store Processed Data** – Save transformed data in Amazon S3, Redshift, or other destinations.
4. **Query with Athena & Redshift Spectrum** – Run SQL queries on the processed data.

AWS Glue Components

- **AWS Glue Data Catalog** – Stores metadata about data.
- **AWS Glue Crawlers** – Scan and infer schemas from raw data.
- **AWS Glue ETL Jobs** – Run **Apache Spark-based** transformations.
- **AWS Glue Triggers** – Automate workflows based on conditions.
- **AWS Glue Studio** – Visual interface for building ETL pipelines.

Creating AWS Glue Crawlers and Linking to an S3 Folder

A **Crawler** in AWS Glue is a service that **automatically scans data sources, infers the schema, and creates metadata tables in the AWS Glue Data Catalog**. When linked to an S3 folder, the Crawler will inspect all files in that folder, extract schema information, and store it as a table in the **AWS Glue Data Catalog**, making it queryable.

What Exactly Does the Crawler Do?

When the AWS Glue Crawler runs, it performs the following tasks:

1. **Scans the S3 Folder**
    - It **recursively** scans all files inside the specified folder.

2. **Infers the Schema**
    - The Crawler detects the data format (CSV, JSON, Parquet, Avro, etc.).
    - It infers column names, data types, and partitions (if applicable).

3. **Creates or Updates a Table in the Glue Data Catalog**
    - The Crawler **registers the schema** in the AWS Glue Data Catalog.
    - It updates the table if new files or schema changes are detected.

4. **Enables Querying with Athena, Redshift Spectrum, or Spark**
    - The table is now accessible via **Amazon Athena (SQL queries)**.
    - Redshift Spectrum or Spark can use this metadata for analytics.

Amazon Athena

Amazon Athena is a **serverless, interactive query service** that allows you to run **SQL queries** on data stored in **Amazon S3** without needing a database or infrastructure.

Key Features of Athena

- **Serverless:** No need to provision or manage infrastructure.
- **Uses Standard SQL:** Supports ANSI SQL for querying data.
- **Works directly with Amazon S3:** No need to load data into a database.
- **Supports Multiple File Formats:** Works with CSV, JSON, Parquet, ORC, and Avro.
- **Integrates with AWS Glue:** Uses Glue Data Catalog for schema management.

How Amazon Athena Works

1. **Data is stored in S3** in raw format (CSV, JSON, Parquet, etc.).
2. **AWS Glue Crawler** scans the data and creates a schema in the **Glue Data Catalog**.
3. **Athena uses the schema** to run **SQL queries** directly on the S3 data.
4. **Query results are stored in S3** and can be visualized in QuickSight.

Amazon QuickSight

Amazon QuickSight is a **business intelligence (BI) service** that enables you to **create, analyze, and share interactive dashboards and visualizations** using AWS data sources like S3, Redshift, Athena, and more.

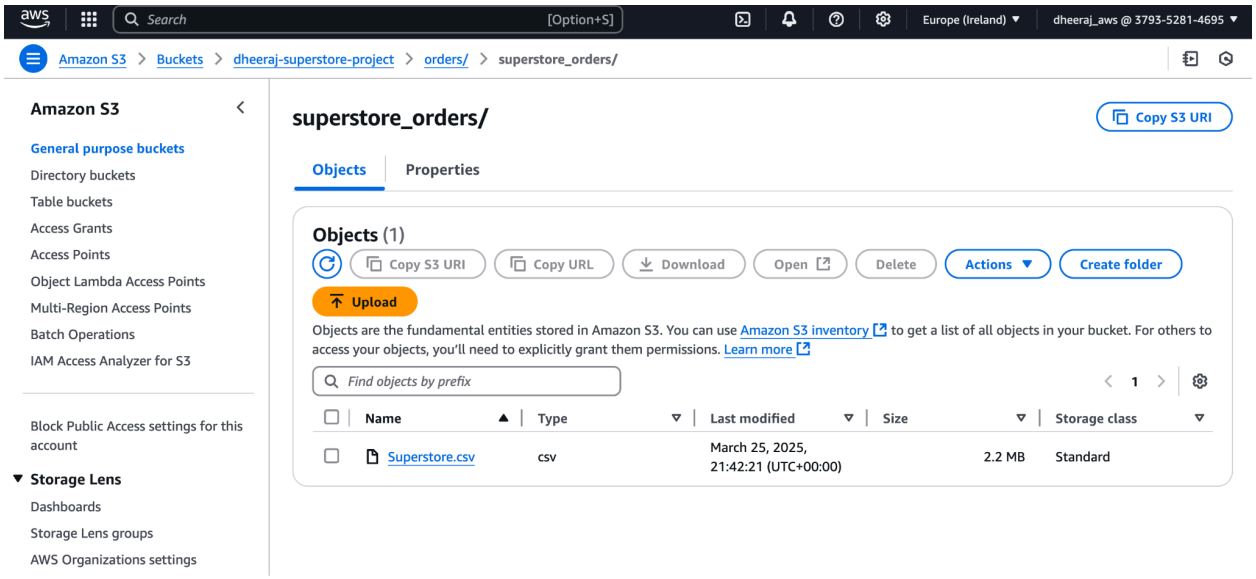
Key Features of Amazon QuickSight

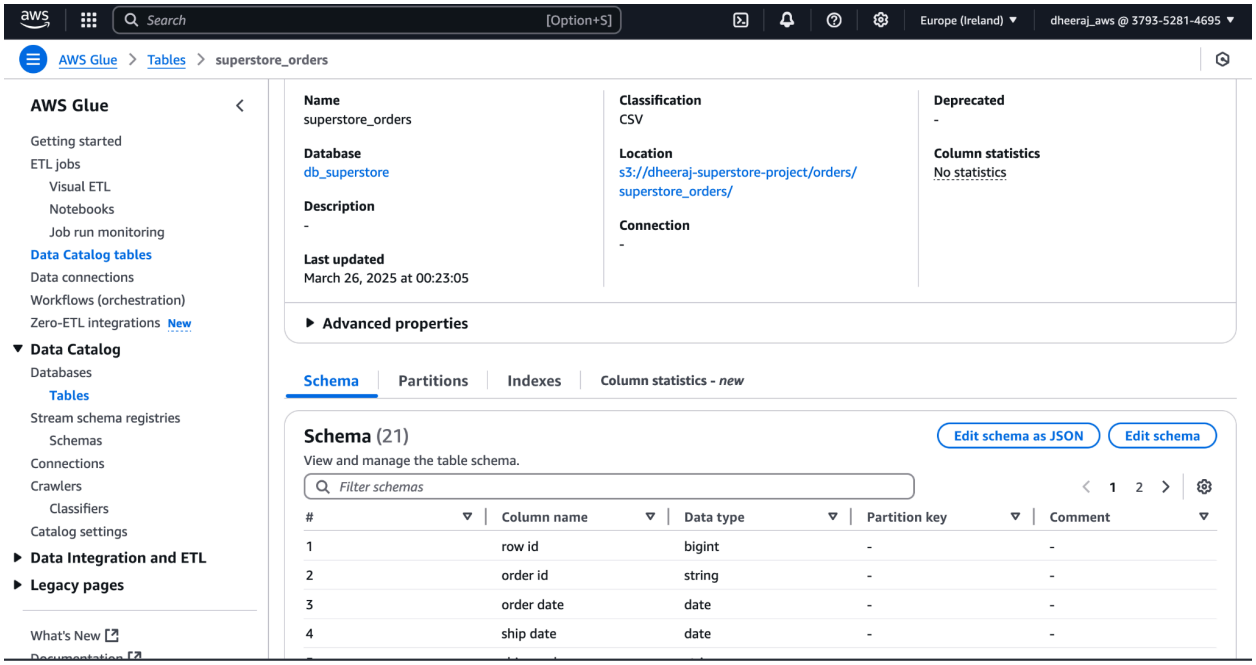
- **Serverless & Scalable:** No infrastructure management, scales automatically.
- **Supports Multiple Data Sources:** AWS services (S3, Redshift, RDS, Athena), and databases (MySQL, PostgreSQL).
- **Pay-Per-Session Pricing:** Cost-effective compared to traditional BI tools.
- **Machine Learning Insights:** Built-in ML-powered anomaly detection and forecasting.
- **SPICE Engine:** In-memory caching for faster performance.
- **Embedded Analytics:** Integrate QuickSight dashboards into apps or websites.

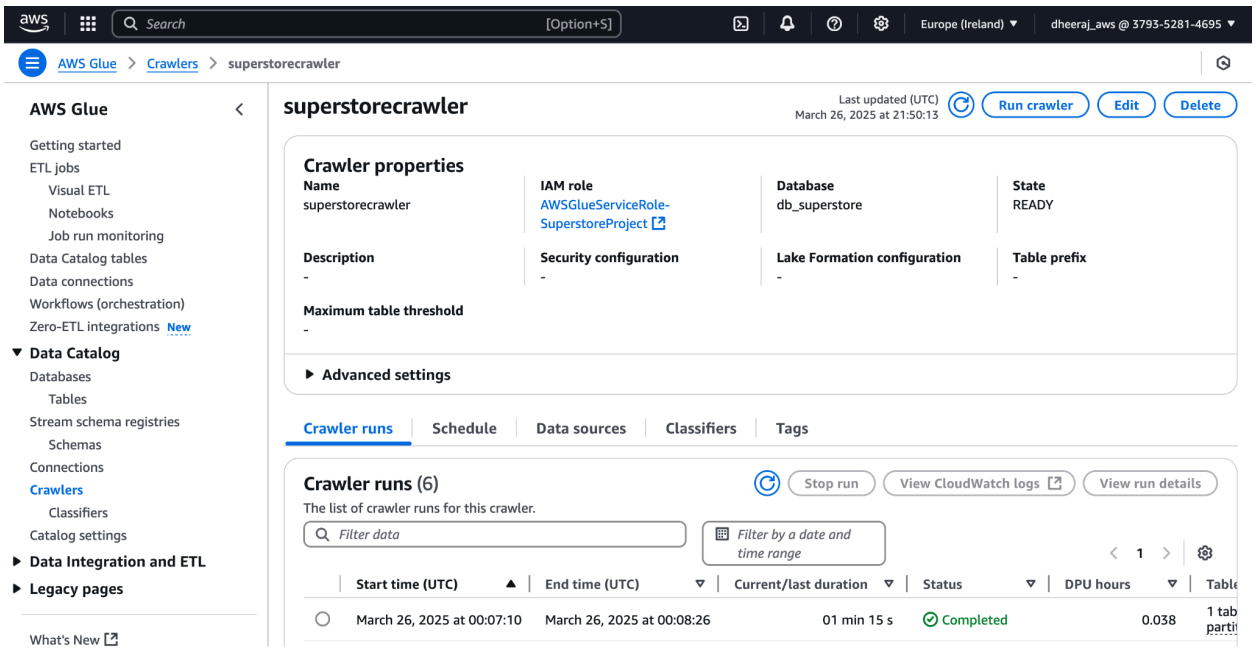
Connecting QuickSight to S3 Using Athena

1. **Store data in Amazon S3** (CSV, JSON, or Parquet format).
2. **Create an AWS Glue Crawler** to catalog the data.
3. **Use AWS Athena** to query the data.
4. **Connect Athena to QuickSight** as a data source.
5. **Build visualizations** using Athena tables.

Screen captures AWS S3 bucket, Crawler, and Data catalog table






Search
[Option+S]
Europe (Ireland)
dheeraj_aws @ 3793-5281-4695
AWS Glue > Crawlers > superstorecrawler
AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
Zero-ETL integrations New
Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
Legacy pages
What's New
superstorecrawler
Last updated (UTC)
March 26, 2025 at 21:50:13
Run crawler
Edit
Delete
Crawler properties
Name
superstorecrawler
IAM role
AWSGlueServiceRole-SuperstoreProject
Database
db_superstore
State
READY
Description
-
Security configuration
-
Lake Formation configuration
-
Table prefix
-
Maximum table threshold
-
Advanced settings
Crawler runs
Schedule
Data sources
Classifiers
Tags
Crawler runs (6)
The list of crawler runs for this crawler.
Stop run
View CloudWatch logs
View run details
Filter data
Filter by a date and time range
Start time (UTC)
End time (UTC)
Current/last duration
Status
DPU hours
March 26, 2025 at 00:07:10
March 26, 2025 at 00:08:26
01 min 15 s
Completed
0.038

SQL queries to analyze the data and generate insights

Total Sales Revenue

```
SELECT SUM(sales) AS total_revenue
FROM "db_superstore"."superstore_orders";
```

Total Number of Orders

```
SELECT COUNT(*) AS total_orders
FROM "db_superstore"."superstore_orders";
```

Most Profitable Product Categories

```
SELECT category, ROUND(SUM(profit),2) AS total_profit
FROM "db_superstore"."superstore_orders"
GROUP BY category
ORDER BY total_profit DESC;
```

Profitability by Region

```
SELECT region, ROUND(SUM(profit),2) AS total_profit
FROM "db_superstore"."superstore_orders"
GROUP BY region
ORDER BY total_profit DESC;
```

Top 5 Best-Selling Products

```
SELECT "product name", SUM(sales) AS total_sales
FROM "db_superstore"."superstore_orders"
GROUP BY "product name"
ORDER BY total_sales DESC
LIMIT 5;
```

Worst 5 Selling Products

```
SELECT  "product name", SUM(sales) AS total_sales
FROM "db_superstore"."superstore_orders"
GROUP BY "product name"
ORDER BY total_sales ASC
LIMIT 5;
```

Average Order Value (AOV)

```
SELECT SUM(sales) / COUNT(DISTINCT "order id") AS avg_order_value
FROM "db_superstore"."superstore_orders";
```

Total Unique Customers

```
SELECT COUNT(DISTINCT "customer id") AS total_customers
FROM "db_superstore"."superstore_orders" ;
```

Screen captures of some queries and results in AWS