**Team 2**

**AWS Data Analytic Platform for the City of Vancouver**

Sree Charan Addala (2316748)

University Canada West

Phase 1

**Dataset** – Employee remuneration and expenses

# **Introduction**

To make City decisions better, Vancouver needs to use mature data analytics which improve how it operates and meets community demands. Greater amounts of recorded data from a city are good for data-driven choices made by its leaders. The information needed includes items like property taxes, information about traffic control, city safety and how the city is planned. By adopting a citywide analytics model, Chicago has become better able to address major social and economic problems for its citizens.

# **Steps for designing and implementing the platform.**

1. Data Ingestion: Collecting raw data from multiple sources, such as government databases. AWS Glue is used for ETL (Extract, Transform, Load) processes.
2. Data storage: Securely store structured and unstructured data in Amazon S3(data lake).
3. Data processing and analysis: Using AWS Glue for cleaning and transforming the data.
4. Data visualization and reporting: Presenting findings through dashboards and reports to aid decision-making.

# **DAP Design and Implementation (Employee remuneration and expenses - Sree Charan Addala)**

## **Descriptive Analysis**

City of Vancouver publishes thorough reports on employee pay and costs to show everyone how public servants are compensated. Because the data is from many years and includes extensive financial information, AWS cloud computing allows for flexible and economical storage, processing and analysis. I studied data about Civil Engineer I employees to learn how their pay and work expenses have gone up or down over time. With Amazon S3, Glue, Athena, I was able to view and analyze these trends using useful metrics. This made it possible for me to notice annual changes, check if the budgeting stayed steady and analyze the alterations in public spending over time and role changes.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 1**

*Data Analytic Platform*

A screenshot of a computer

AI-generated content may be incorrect.

Employee Remuneration and Expense draw.io shows how data and services flow and are refined in the system. It includes a Data Analytics Platform (DAP).

## **Step 1: Data Ingestion**

**Figure 2**

*S3 raw bucket creation*

A screenshot of a computer

AI-generated content may be incorrect.

*Note*. Source: (Self).

The data used for Employee Remuneration and Expenses is from the City of Vancouver Open Data Portal. The data collection process is done, and the raw information is now kept in the S3 bucket called employeeremuneration -raw-sca. This is the first essential task in the pipeline which prepares the data for more steps like transformation, enrichment and analysis.

**Figure 3**

*Data ingestion from City of Vancouver*

*A screenshot of a computer

AI-generated content may be incorrect.*

*Note*. Source: (Self).

This image shows that the Employee Remuneration and Expenses data from the City of Vancouver Open Data Portal has been successfully put into an Amazon S3 bucket. The file has been uploaded in .csv format and stored where it can be accessed further for analysis.

**Step 2: Data Profiling**

**Figure 4**

*Data Profile with Job creation*

A screenshot of a computer

AI-generated content may be incorrect.

*Note*. Source: (Self).

Creating an AWS Glue job helped profiling the data in the screenshot. It stores the data from the Employee Remuneration and Expenses raw bucket (employeeremuneration-raw-sca) as the transformed bucket (employeeremuneration-cln-sca) to make sure the original data remains unchanged.

## **Step 3: Data Cleaning**

**Figure 5**

*Applying cleaning job for the dataset*

A screenshot of a computer

AI-generated content may be incorrect.

*Note*. Source: (Self)

Cleaning the data before using it to work with the dataset. Few data cleaning techniques are applied to remove blanks and rename columns, change data types, array to string and integer as needed. The dataset is cleaned and the required files are made after finishing the cleaning Job.

**Figure 6**

*User Dataset (cleaned)*

A screenshot of a computer

AI-generated content may be incorrect.

*Note.* Source: (Self)

Data has been cleaned for User. Cleaning must be done successfully for this to happen.

**Figure 7**

*System Dataset (cleaned)*

A screenshot of a computer

AI-generated content may be incorrect.

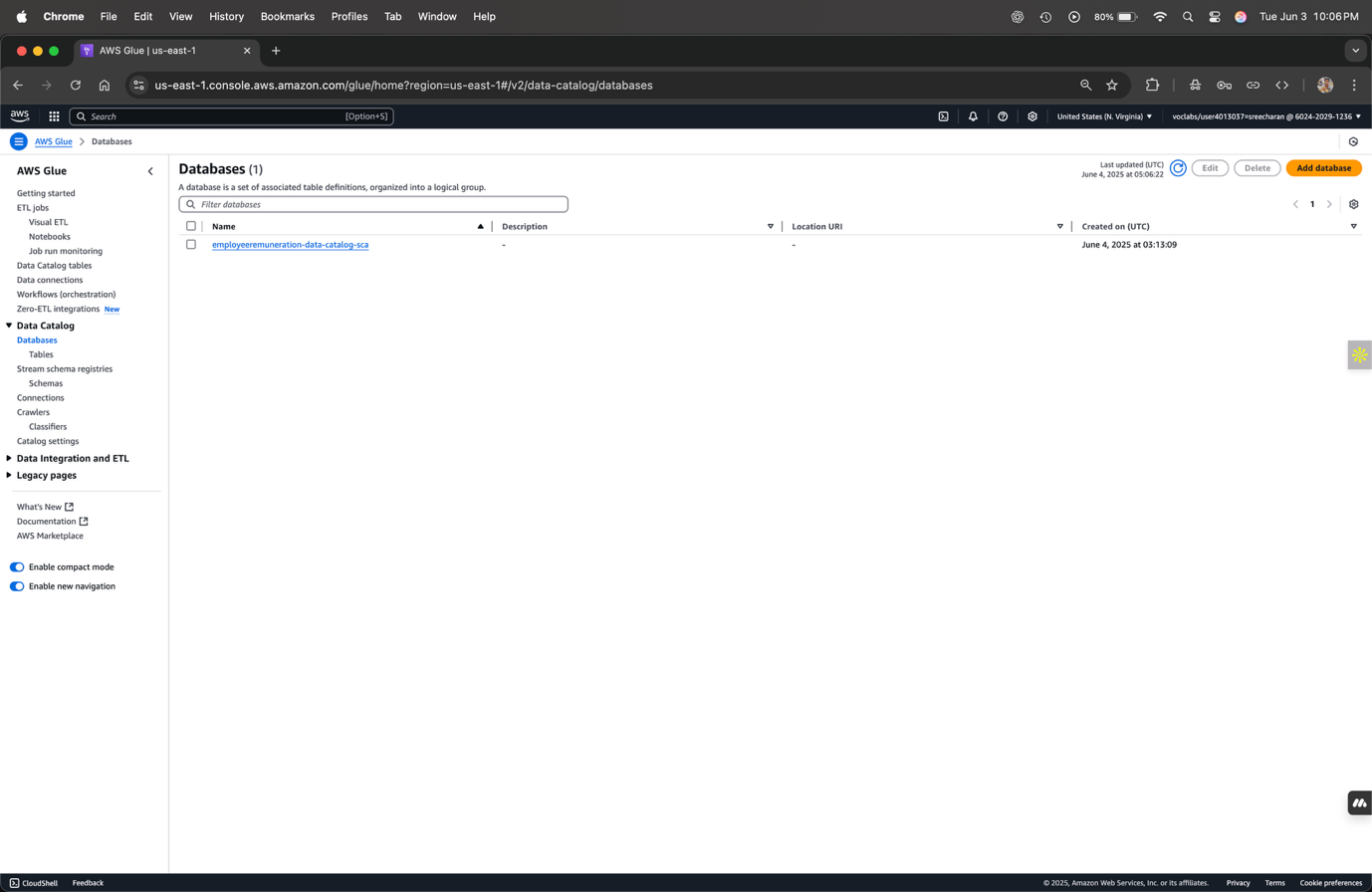
*Note.* Source: (Self)

After data cleaning, everything looked good in System. The purpose is made possible when a proper cleaning job is executed.

## **Step 4: Data Cataloging**

**Figure 8**

*Data Catalogue Creation in AWS*

**

*Note.* Source: (Self)

Here, we can see how a data catalogue has been successfully made. It is built with tables based on the key metrics.

**Step 5: Data Summarization**

**Figure 9**

*Creation of Visual ETL Jobs*

A screenshot of a computer

AI-generated content may be incorrect.

*Note.* Source: (Self)

To process data for the Employee Remuneration and Expenses dataset, Visual ETL Jobs were made using AWS Glue. Transforming and gathering key data for the business was done using the visual ETL service in AWS Glue. Applying aggregate functions, filters and sorting by year was important to get an average for the pay and expenses of Civil Engineer I employees. I could preview the results at every transformation level which confirmed their accuracy and consistency. The final dataset in summary form was saved to the S3 bucket specified by the user as well as for future automation use.

**Cost Estimation**

**Figure 13**

*AWS Pricing Estimation*

A screenshot of a computer

AI-generated content may be incorrect.

*Note.* Source: (Self)

The AWS Pricing Calculator was used to estimate the cost of processing the Employee Remuneration and Expenses dataset based on the usage of services such as Amazon S3 and AWS Glue ETL Jobs. The calculation considered the number of Data Processing Units (DPUs) allocated for the job. Higher DPU allocation results in faster data processing but also increases the overall cost, helping balance performance and budget requirements effectively.