Project Part 02 – Team Report

AWS Data Analytic Platform for The City of Vancouver

"Phase 2"

University Canada West

BUSI 653(ON-FALL24-01)

Professor: Mahmood Mortazavi Dehkordi

Barot Raj Dineshbhai (2304564)

Joud Sabri Asfour (2212611)

Aigbe Umole (1510627)

Samuel Omotayo (2302001)

December 8, 2024

DAP Implementation, Individual work Team member 1(Asfour, Joud Sabri Jehad)

1. Enrichment of data:

- Add any derived or necessary columns if it is applicable.
- Enrichment of data with the appropriate methods for the identification of missing or incomplete entries.

2. Protection of data:

- Mask sensitive data (e.g. personal identifiable information) through either pseudonymization or hashing.
- For sensitive column, apply encryption if necessary.

3. Governance of data:

- Consistent data types, naming conventions, and schema integrity are ensured.
- Validate against schema rules.

4. Observability of Data:

- Assure of a range of metrics such as missing values, duplicates, or outliers.
- Generate visualizations for monitoring.

There are 29 columns, and these columns have different data types, such as numerical, categorical and textual. How to get these steps done will be as follows:1. Data Enrichment

1. Adding constructed columns:

- Calculate percentage change in land and improvement values from previous to current values.
- Calculate total property value which is the sum of land and improvements.

2.Data Protection

- Masks sensitive data:
- Pseudonymization of PID and PROPERTY POSTAL CODE.
- Hashed columns (e.g. FOLIO) for anonymization.

3. Data Governance

- Make sure schema valid:
- Convert all columns to datatype appropriate when needed.
- Rename columns to follow
- Data Observability
- Investigate data quality:
- Missing and duplicate entries.
- Outliers in numeric fields such as CURRENT LAND VALUE.
- Generate illustrations of data and outlying portion.

The process of data transformation is finished. Here are the results: Summary of Observations

1. **Missing Values**:
    a. narrative_legal_line3 has 1 missing value.
    b. Derived column improvement_value_change_% has 1 missing value due to division by zero.
2. **Duplicates**:
    - No duplicates were found.
3. **Outliers in current_land_value**:
    - 30 outliers detected based on the interquartile range (IQR).

Processed Dataset

- Enriched with percentage changes and total property value.
- Sensitive columns (PID, PROPERTY_POSTAL_CODE, and FOLIO) are pseudonymized or hashed.
- Columns are standardized for consistency.

```
import pandas as pd
import hashlib
import numpy as np
```

```
# Load the dataset
file_path = "proj-cln-joud_2024_11_20.csv"  # Replace
with the actual file path
data = pd.read_csv(file_path)
```

1. Data Enrichment

Data enrichment involves adding new derived features to make the dataset more informative or useful:

- **LAND_VALUE_CHANGE_%**: This derived column calculates the percentage change in land value (CURRENT_LAND_VALUE vs. PREVIOUS_LAND_VALUE), providing insights into property value trends.
- **IMPROVEMENT_VALUE_CHANGE_%**: This column computes the percentage change in improvement value (CURRENT_IMPROVEMENT_VALUE vs. PREVIOUS_IMPROVEMENT_VALUE), giving additional context to property development.
- **TOTAL_PROPERTY_VALUE**: A composite feature representing the sum of land and improvement values, summarizing a property's total worth.

By adding these columns, the dataset becomes more valuable for analytics and decision-making.

```
# Step 1: Data Enrichment
# Add derived columns
data['LAND_VALUE_CHANGE_%'] =
((data['CURRENT_LAND_VALUE'] -
data['PREVIOUS_LAND_VALUE']) /
```

```
data['PREVIOUS_LAND_VALUE']) * 100
data['IMPROVEMENT_VALUE_CHANGE_%'] =
((data['CURRENT_IMPROVEMENT_VALUE'] -
data['PREVIOUS_IMPROVEMENT_VALUE']) /
```

```
data['PREVIOUS_IMPROVEMENT_VALUE']) * 100
data['TOTAL_PROPERTY_VALUE'] =
data['CURRENT_LAND_VALUE'] +
data['CURRENT_IMPROVEMENT_VALUE']
```

2. Data Protection

Data protection ensures the confidentiality and security of sensitive information:

- **Pseudonymization**:
    - PID and PROPERTY_POSTAL_CODE are pseudonymized by applying a **SHA-256 hash function**, replacing original values with irreversible hashed values. This protects sensitive identifiers while preserving uniqueness for analytical purposes.
- **Hashing**:
    - FOLIO is also hashed with SHA-256, anonymizing it while maintaining consistent hashing for identical values.

This approach adheres to data privacy regulations like GDPR by making sensitive data unusable for unauthorized parties.

```
# Step 2: Data Protection
# Pseudonymize PID and PROPERTY_POSTAL_CODE
data['PID_Pseudonym'] = data['PID'].apply(lambda x:
hashlib.sha256(x.encode()).hexdigest())
data['POSTAL_CODE_Pseudonym'] =
data['PROPERTY_POSTAL_CODE'].apply(lambda x:
hashlib.sha256(x.encode()).hexdigest())
```

```
# Hash FOLIO
data['FOLIO_Hashed'] = data['FOLIO'].apply(lambda x:
hashlib.sha256(str(x).encode()).hexdigest())
```

3. Data Governance

Data governance ensures consistency, structure, and adherence to standards within the dataset:

- **Column Standardization**:
    - Column names are converted to lowercase and spaces are replaced with underscores (_) for uniformity and to ensure compatibility with coding conventions.
- **Schema Integrity**:
    - Data processing ensures derived and transformed columns fit into the schema, maintaining the overall consistency of the dataset.

Such practices improve the dataset's usability and make it easier to share or integrate with other systems.

```
# Step 3: Data Governance
# Standardize column names
data.rename(columns=lambda x:
x.strip().lower().replace(' ', '_'), inplace=True)
```

```
# Step 3: Data Governance
# Standardize column names
data.rename(columns=lambda x:
x.strip().lower().replace(' ', '_'), inplace=True)

# Step 4: Data Observability
```

```
# Step 3: Data Governance
# Standardize column names
data.rename(columns=lambda x:
x.strip().lower().replace(' ', '_'), inplace=True)
```

4. Data Observability

Data observability focuses on monitoring and maintaining data quality:

- **Missing Values**:
    - The code calculates the count of missing values in each column using isnull().sum() and includes this information in the observation's summary.
- **Duplicate Detection**:
    - The code checks for duplicate rows using data.duplicated().sum() to identify redundant records that might skew analysis.
- **Outlier Detection**:
    - The code flags outliers in the current_land_value column based on the interquartile range (IQR) method:
        - Thresholds are calculated using Q1 - 1.5 * IQR and Q3 + 1.5 * IQR.

- Records falling outside this range are flagged as outliers in the land_value_outlier column.

```
# Step 4: Data Observability
# Detect missing values and duplicates
missing_values = data.isnull().sum()
duplicates = data.duplicated().sum()
```

```
# Flag outliers in CURRENT_LAND_VALUE
q1 = data['current_land_value'].quantile(0.25)
q3 = data['current_land_value'].quantile(0.75)
iqr = q3 - q1
outlier_threshold_low = q1 - 1.5 * iqr
outlier_threshold_high = q3 + 1.5 * iqr
data['land_value_outlier'] =
~data['current_land_value'].between(outlier_threshold
_low, outlier_threshold_high)
```

```
# Observability summary
observations = {
    "missing_values": missing_values,
    "duplicates": duplicates,
    "land_value_outliers":
data['land_value_outlier'].sum()
}
```

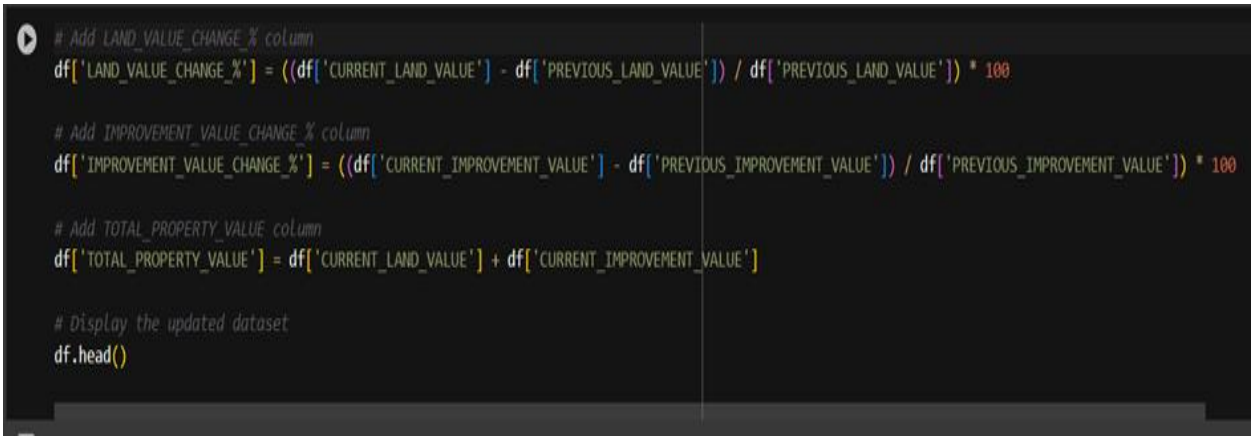
```
# Save the processed dataset
processed_file_path = "processed_dataset.csv"  #
Replace with your desired output file path
data.to_csv(processed_file_path, index=False)
```

```
# Print summary of observations
print("Observations:")
for key, value in observations.items():
    print(f"{key}: {value}")
```

```


print(f"\nProcessed dataset saved at:
{processed_file_path}")
```

These steps provide a comprehensive view of data quality, helping identify and address potential issues.

Step 5: Data Enrichment



```
# Add LAND_VALUE_CHANGE_% column
df['LAND_VALUE_CHANGE_%'] = ((df['CURRENT_LAND_VALUE'] - df['PREVIOUS_LAND_VALUE']) / df['PREVIOUS_LAND_VALUE']) * 100

# Add IMPROVEMENT_VALUE_CHANGE_% column
df['IMPROVEMENT_VALUE_CHANGE_%'] = ((df['CURRENT_IMPROVEMENT_VALUE'] - df['PREVIOUS_IMPROVEMENT_VALUE']) / df['PREVIOUS_IMPROVEMENT_VALUE']) * 100

# Add TOTAL_PROPERTY_VALUE column
df['TOTAL_PROPERTY_VALUE'] = df['CURRENT_LAND_VALUE'] + df['CURRENT_IMPROVEMENT_VALUE']

# Display the updated dataset
df.head()
```

- **Objective:** Add derived features to enhance the dataset's value for analysis.
- **Steps:**
    - Calculate percentage changes:
        - LAND_VALUE_CHANGE_%: ((CURRENT_LAND_VALUE - PREVIOUS_LAND_VALUE) / PREVIOUS_LAND_VALUE) * 100.
    - IMPROVEMENT_VALUE_CHANGE_%: ((CURRENT_IMPROVEMENT_VALUE - PREVIOUS_IMPROVEMENT_VALUE) / PREVIOUS_IMPROVEMENT_VALUE) * 100. Compute TOTAL_PROPERTY_VALUE: Sum of CURRENT_LAND_VALUE and CURRENT_IMPROVEMENT_VALUE.
    - Handle missing values in derived columns (e.g., replace NaN with 0).
    - Save the enriched dataset as enriched_property_data.csv.

Step 6: Data Protection

```
# Apply SHA-256 hashing for pseudonymization of PID
df['HASHED_PID'] = df['PID'].apply(lambda x: hashlib.sha256(x.encode()).hexdigest())

# Apply SHA-256 hashing for pseudonymization of PROPERTY_POSTAL_CODE
df['HASHED_POSTAL_CODE'] = df['PROPERTY_POSTAL_CODE'].apply(lambda x: hashlib.sha256(x.encode()).hexdigest())

# Display the dataset with hashed columns
df[['PID', 'HASHED_PID', 'PROPERTY_POSTAL_CODE', 'HASHED_POSTAL_CODE']].head()
```

	PID	HASHED_PID	PROPERTY_POSTAL_CODE	HASHED_POSTAL_CODE
0	PID00001	794f9052c465c039ccdcdd3c427ec4580e11c77a8a4ee8...	V3A12Z4	b6e690042e0c735f1d0d79c5fe0dcedd8960dce81ee14f...
1	PID00002	b74f52baf8c09118356dcff03a118e04f4924b059c9be1...	V1A50Z7	595810e3c49351ccaac3a71ea417082c5daca21884bece...
2	PID00003	64452761 92eb60f66ac5744309c42a09f74baa2381b23e...	V6A54Z7	a561a92f5c382012f88e9cc6f65115a9410742253ce4e8...
3	PID00004	bad3c96d8a438b1f489568b135f6644cee0fbc8c165ee3...	V3A60Z2	0768a6f8c5001536a4239b246fe5642c5e9c93a6133b5d...
4	PID00005	ea0345e7ef4be96907dc7f12ec8dec5503fe68e8ab735a...	V8A67Z3	78ed46abeec74313547931338b2088c6a29f81d482f706...

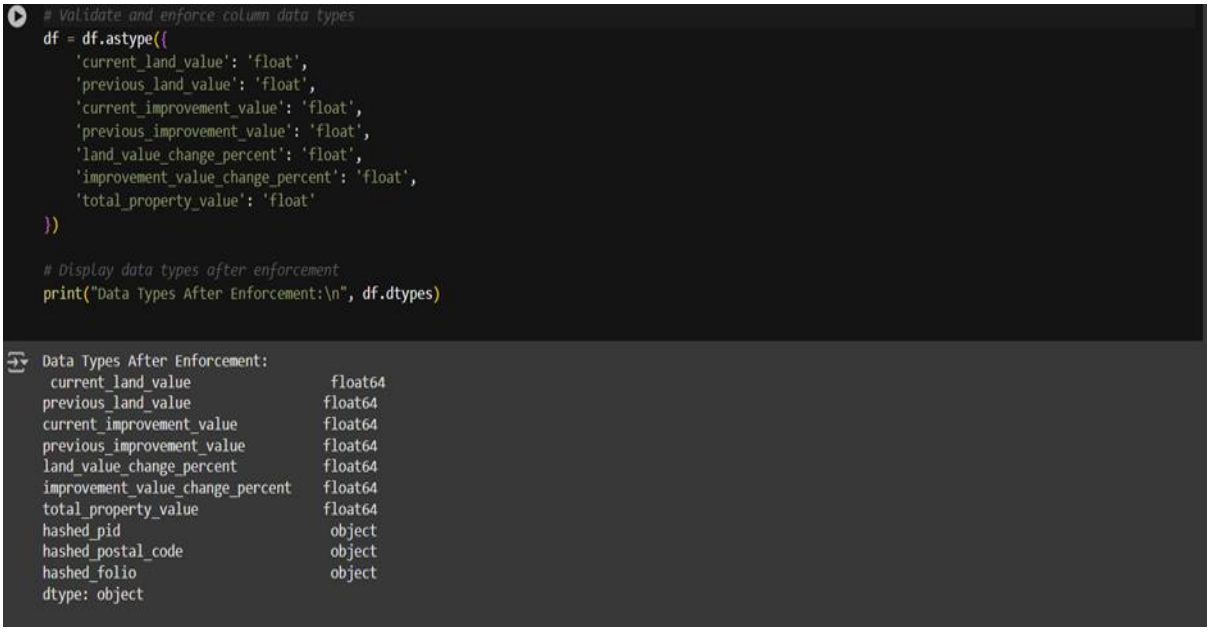
```
# Apply SHA-256 hashing for FOLIO
df['HASHED_FOLIO'] = df['FOLIO'].apply(lambda x: hashlib.sha256(x.encode()).hexdigest())

# Display the original and hashed FOLIO columns
df[['FOLIO', 'HASHED_FOLIO']].head()
```

	FOLIO	HASHED_FOLIO
0	F6849349	23b9a1e5dc61cb889f8037446e9902a701f6c8046e73c7...
1	F8407542	db3f34a43c678118ccb70004a3ed27402cf71d90c545ea...
2	F1528497	8ecbec068b1f80cca94b62d448090ddd7c7e0224726fbf...
3	F5359589	b0ea1bbc5ae1119e718108da5f4008bd044292645466f1...
4	F2659665	ff6fbaf2984ea33a22f1a951ba636fa47e4a3b8d769204...

- **Objective:** Secure sensitive information while maintaining data usability.
- **Steps:**
    - **Pseudonymization:**
        - Apply SHA-256 hashing on PID and PROPERTY_POSTAL_CODE.
    - **Hashing:**
        - Hash FOLIO column with SHA-256 for anonymization.

- Remove original sensitive columns after pseudonymization and hashing.
- Save the protected dataset as protected_property_data.csv.

Step 7: Data Governance



```
# Validate and enforce column data types
df = df.astype({
    'current_land_value': 'float',
    'previous_land_value': 'float',
    'current_improvement_value': 'float',
    'previous_improvement_value': 'float',
    'land_value_change_percent': 'float',
    'improvement_value_change_percent': 'float',
    'total_property_value': 'float'
})

# Display data types after enforcement
print("Data Types After Enforcement:\n", df.dtypes)
```

```
Data Types After Enforcement:
 current_land_value                float64
previous_land_value                float64
current_improvement_value          float64
previous_improvement_value         float64
land_value_change_percent          float64
improvement_value_change_percent   float64
total_property_value               float64
hashed_pid                          object
hashed_postal_code                  object
hashed_folio                        object
dtype: object
```

```python
# Validate and enforce column data types
df = df.astype({
    'current_land_value': 'float',
    'previous_land_value': 'float',
    'current_improvement_value': 'float',
    'previous_improvement_value': 'float',
    'land_value_change_percent': 'float',
    'improvement_value_change_percent': 'float',
    'total_property_value': 'float'
})

# Display data types after enforcement
print("Data Types After Enforcement:\n", df.dtypes)

# Before validating schema integrity, fill NaN values in relevant columns
# This will handle missing values that might have been introduced
# during calculations in the enrichment step.
df['current_land_value'].fillna(df['current_land_value'].mean(), inplace=True)
df['previous_land_value'].fillna(df['previous_land_value'].mean(), inplace=True)
df['total_property_value'].fillna(df['total_property_value'].mean(), inplace=True)
df['land_value_change_percent'].fillna(0, inplace=True) # Or any other appropriate value

# Validate schema integrity
missing_values = df.isnull().sum()
print("Missing Values After Schema Validation:\n", missing_values)

# Confirm no missing or erroneous entries
assert missing_values.sum() == 0, "Schema validation failed! Missing values detected."

print("Schema validation passed successfully!")
```

- **Objective:** Ensure consistency, structure, and schema integrity.
- **Steps:**
    - Standardize column names:
        - Convert to lowercase, replace spaces with underscores, and handle special characters.
    - Enforce consistent data types:
        - Ensure numeric columns (e.g., current_land_value, total_property_value) are of type float.
    - Validate schema integrity:
        - Check for missing or invalid entries and ensure derived columns align with the schema.
    - Save the governed dataset as governed_property_data.csv.

Step 8: Data Observability

```
# Check for missing values
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)

# Visualize missing values
import matplotlib.pyplot as plt

missing_values.plot(kind='bar')
plt.title('Missing Values per Column')
plt.xlabel('Columns')
plt.ylabel('Number of Missing Values')
plt.show()
```

```
Missing Values:
 current_land_value                0
previous_land_value                0
current_improvement_value          0
previous_improvement_value         0
land_value_change_percent          0
improvement_value_change_percent   0
total_property_value               0
hashed_pid                         0
hashed_postal_code                 0
hashed_folio                       0
dtype: int64
```

- **Objective:** Monitor and maintain data quality with metrics and visualizations.
- **Steps:**
    - **Missing Values:**
        - Identify missing values in each column and visualize them (e.g., bar chart).
    - **Duplicate Rows:**
        - Count and optionally remove duplicate rows.
    - **Outlier Detection:**
        - Use the IQR method to detect outliers in numerical columns (e.g., current_land_value).
        - Flag outliers for further analysis.

- **Summary Metrics:**
    - Report missing values, duplicates, and outliers for quality monitoring.
- Save the observed dataset with outlier flags as observed_property_data.csv.

DAP Implementation, Individual work Team member 4(Aigbe James Umole)

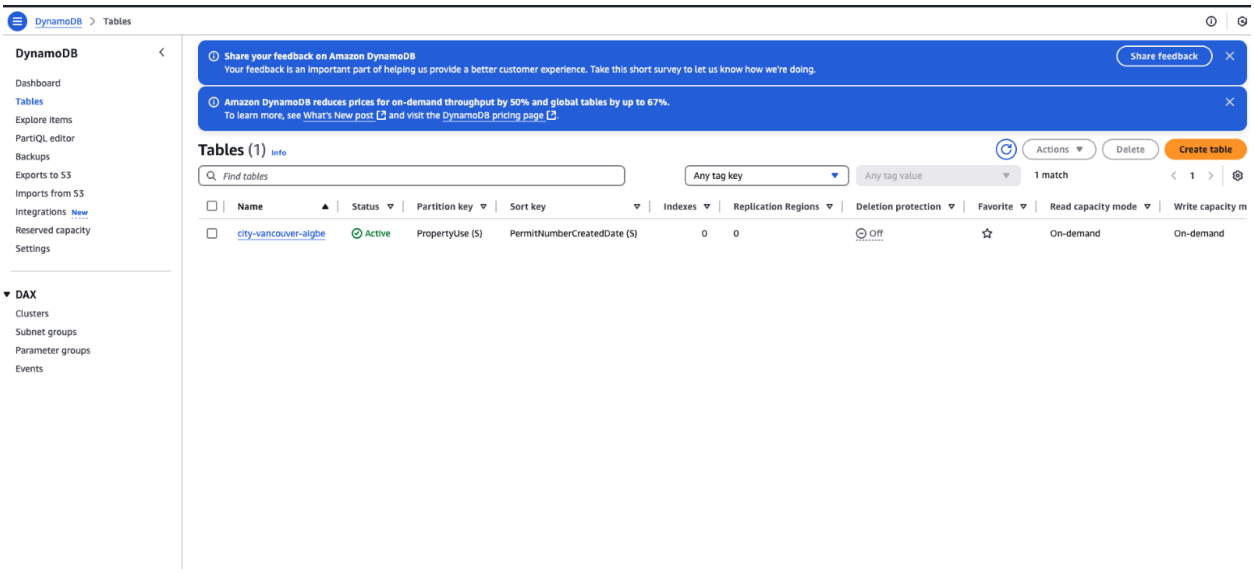
Data enriching:

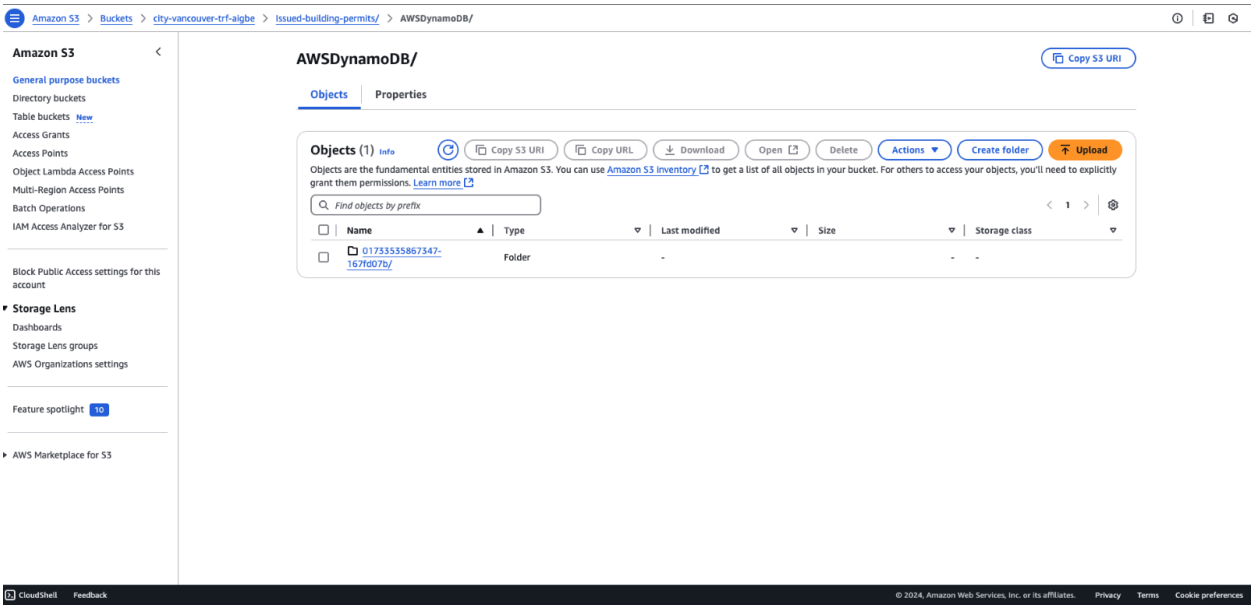
Data enriching refers to the process of enhancing the quality and value of your existing datasets by incorporating additional information from external sources.

To perform data enriching to begin with;

DynamoDB

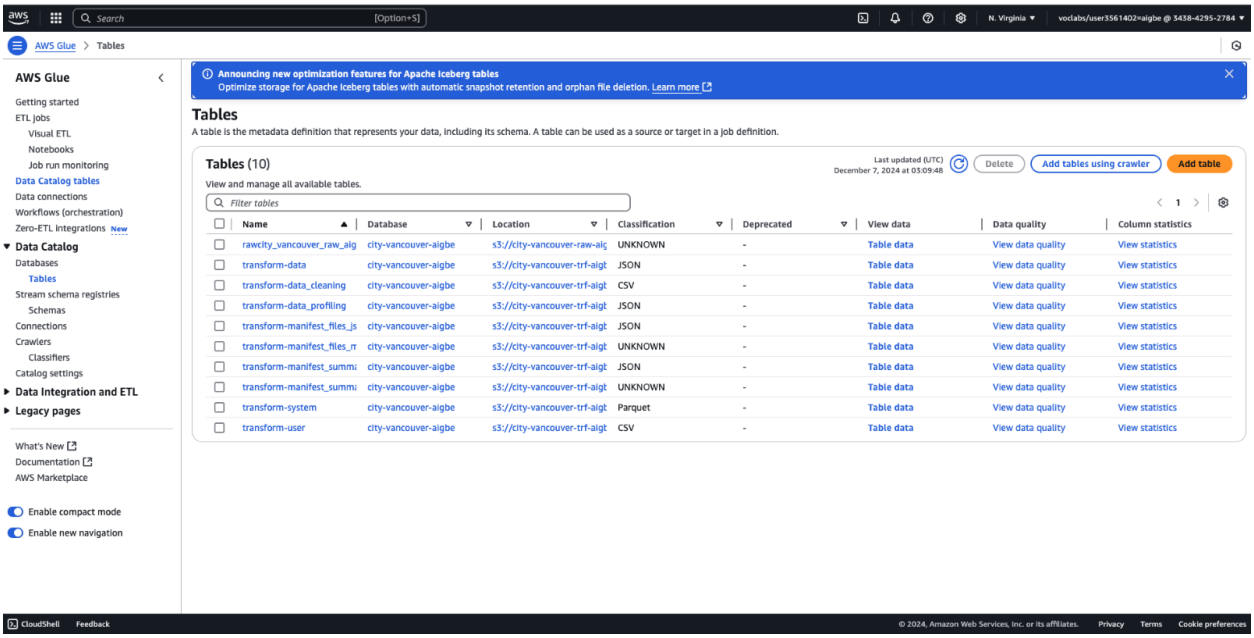
Creating the partitioning for the data set and performing ingestion of Vancouver city data.





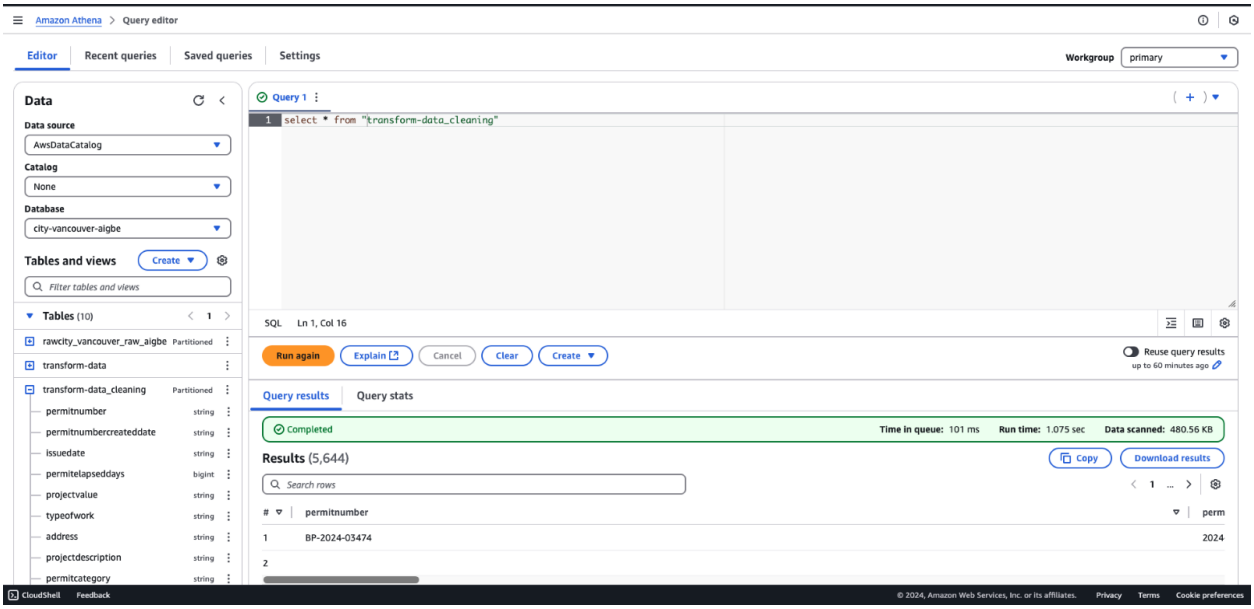
AWS Glue – Crawlers

With the AWS glue data crawlers, I extracted all the data sets I transformed so far.



AWS Athena

With AWS Athena, I was able to query the tables i had created with AWS crawlers.



Data Protection

Data protection in AWS is a comprehensive approach to safeguarding sensitive information stored and processed within the cloud environment. It involves a combination

of technical, organizational, and administrative measures to ensure data confidentiality, integrity, and availability.

Key data protection steps include

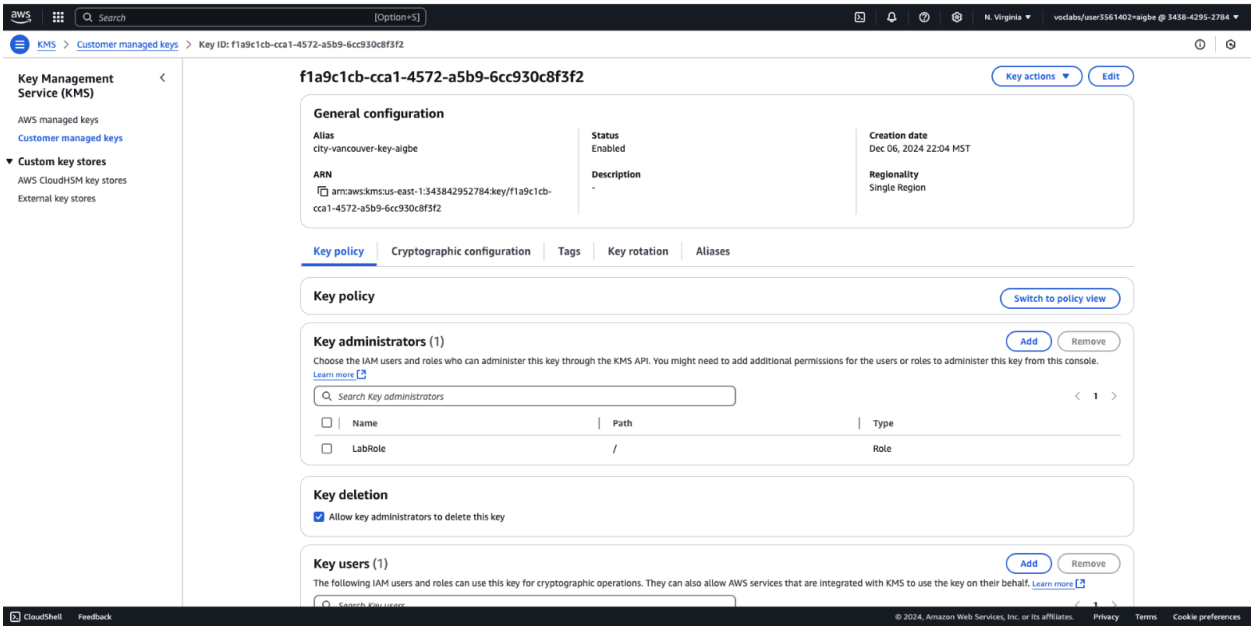
- Encryption
- Access control

Types of Protection:
Confidentiality

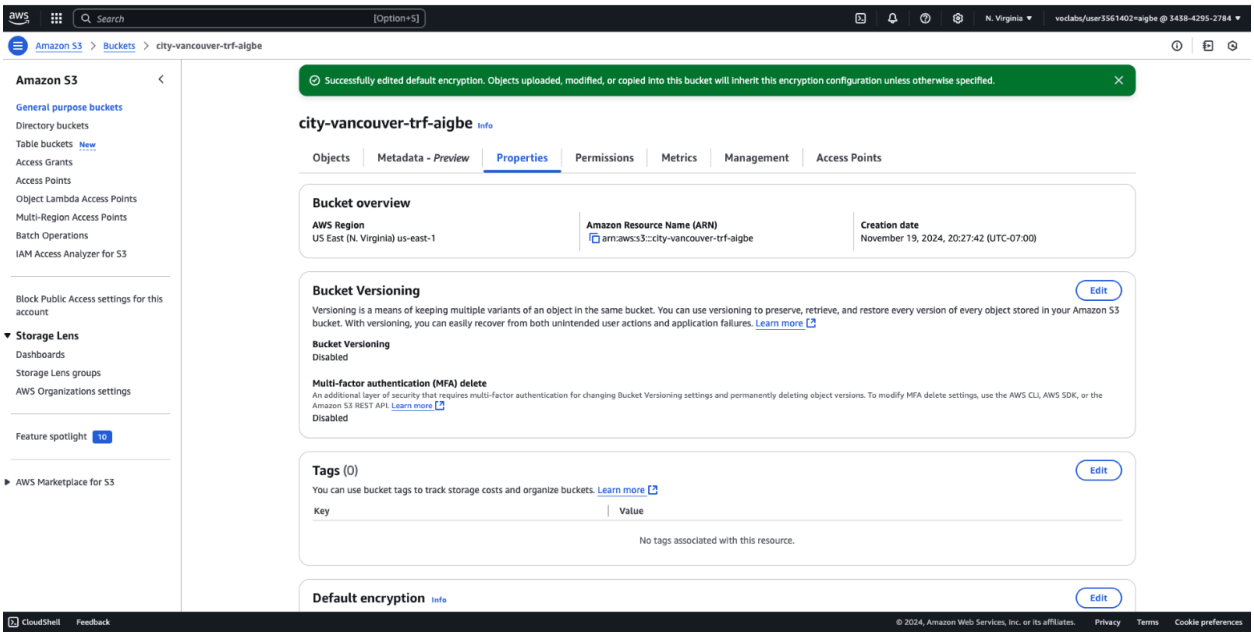
Integrity protection

Availability protection

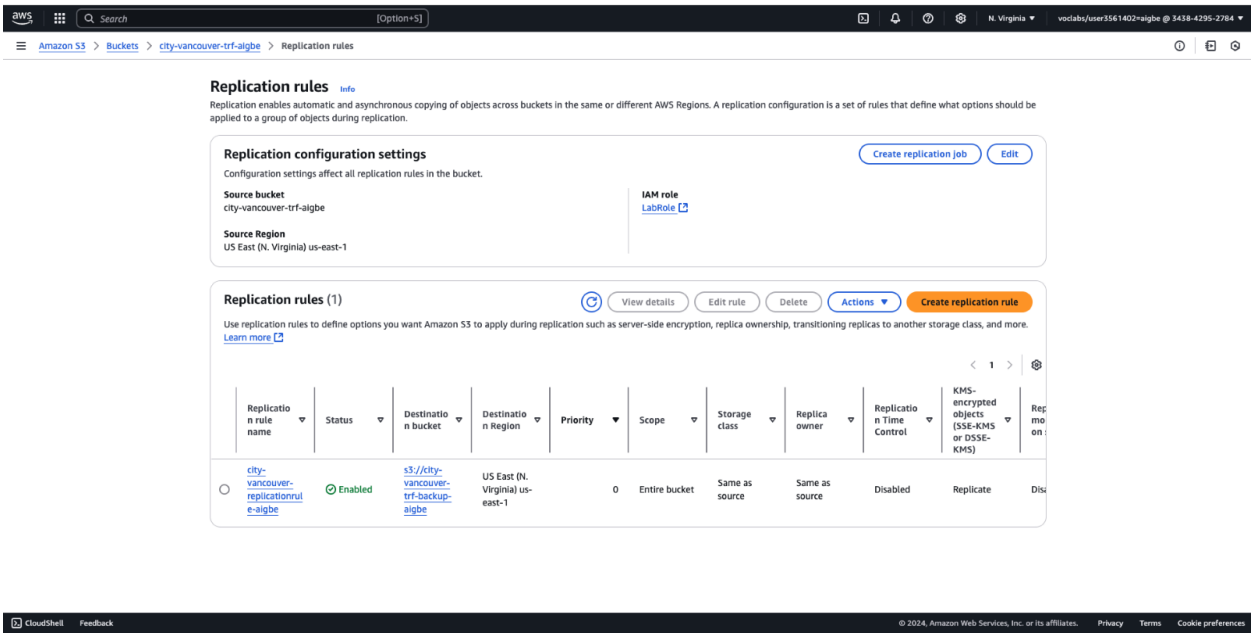
Encryption technique - KMS



Encryption of the S3 bucket for the City of Vancouver data



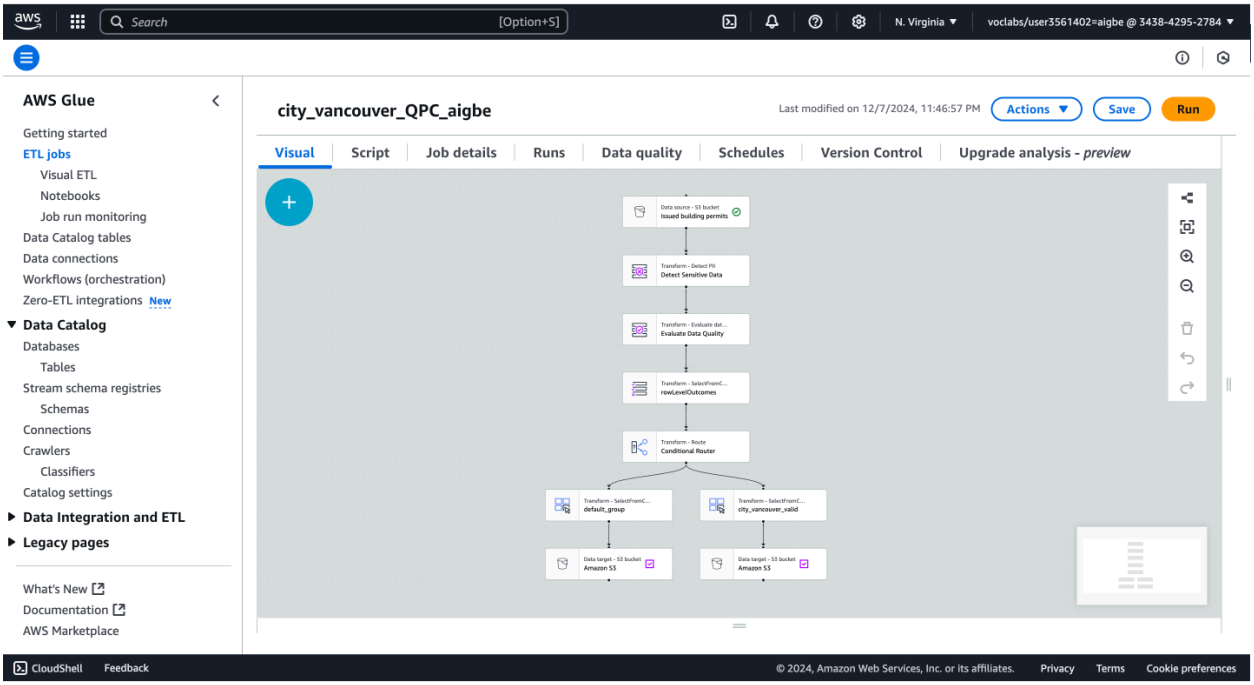
Data backup- replication



Data Governance

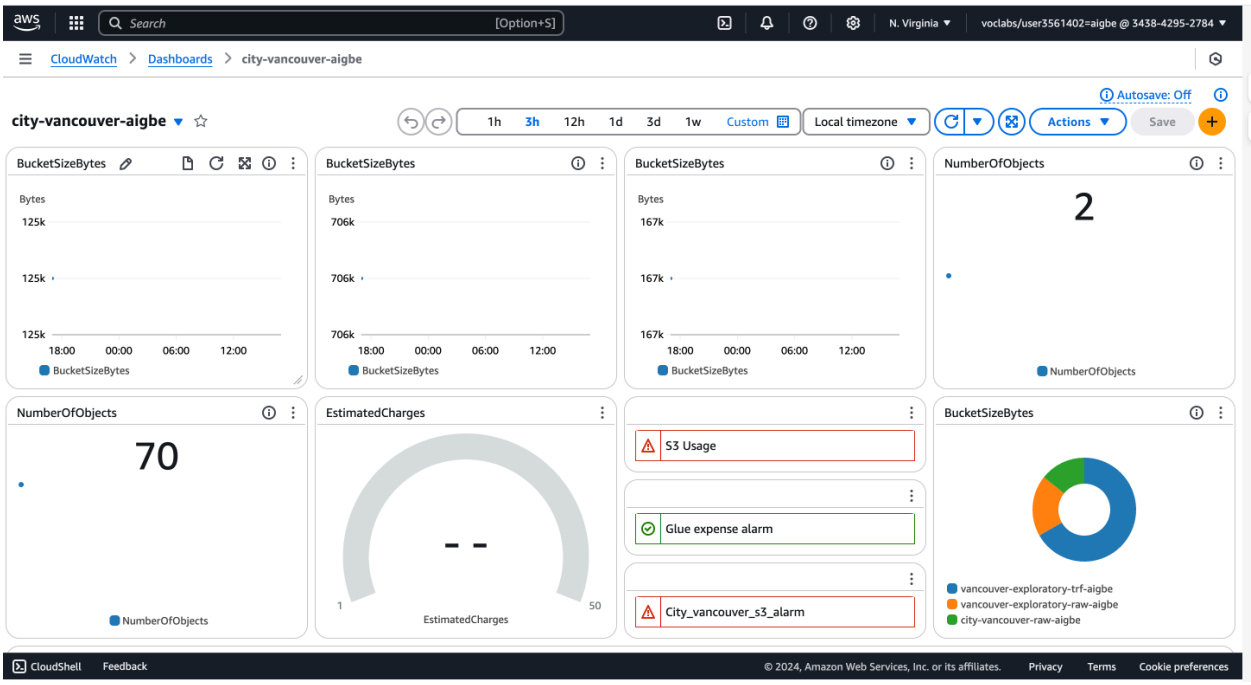
Data governance is a comprehensive framework that ensures the integrity, security, and quality of an organization's data. This includes:

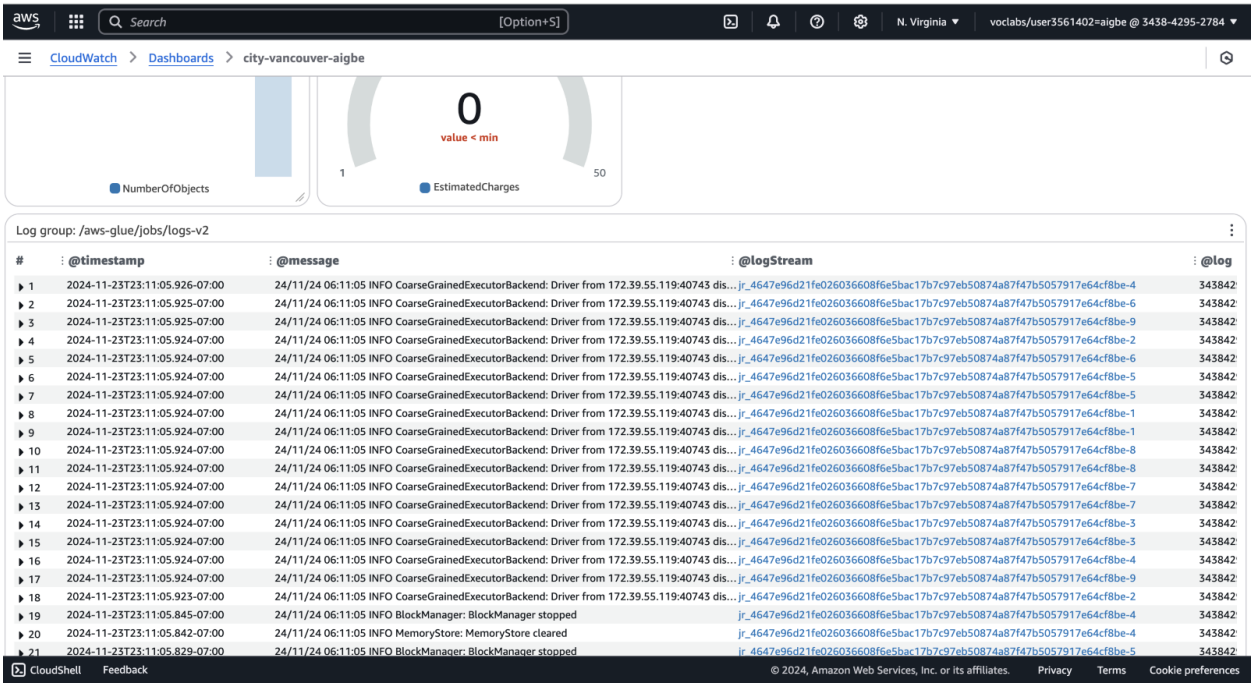
- **Data Quality:** Ensuring data is accurate, complete, consistent, timely, and valid.
- **Data Security:** Protecting data from unauthorized access, use, disclosure, disruption, modification, or destruction.
- **Data Privacy:** Adhering to data privacy regulations and protecting individuals' personal information.
- **Data Compliance:** Ensuring that data practices comply with industry standards and legal requirements.



Data Observability: is the practice of monitoring, managing, and maintaining data to ensure its quality, availability, and reliability.

- **Identify and resolve data issues proactively:** By monitoring data pipelines and systems, organizations can quickly detect and address problems before they impact downstream processes or decision-making.
- **Improve data quality:** Data observability tools can help identify and correct data quality issues, such as missing values, inconsistencies, and errors.
- **Optimize data pipelines:** By analyzing data flow and performance metrics, organizations can identify bottlenecks and optimize data pipelines for efficiency and scalability.




aws
Search
[Option+S]
N. Virginia
voclabs/user3561402=aigbe @ 3438-4295-2784
CloudWatch
Dashboards
city-vancouver-aigbe
NumberOfObjects
0
value < min
1
50
EstimatedCharges
Log group: /aws-glue/jobs/logs-v2
@timestamp @message @logStream @log
1 2024-11-23T23:11:05.926-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-4 343842
2 2024-11-23T23:11:05.925-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-6 343842
3 2024-11-23T23:11:05.925-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-9 343842
4 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-2 343842
5 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-6 343842
6 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-5 343842
7 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-5 343842
8 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-1 343842
9 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-1 343842
10 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-8 343842
11 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-8 343842
12 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-7 343842
13 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-7 343842
14 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-3 343842
15 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-3 343842
16 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-4 343842
17 2024-11-23T23:11:05.924-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-9 343842
18 2024-11-23T23:11:05.923-07:00 24/11/24 06:11:05 INFO CoarseGrainedExecutorBackend: Driver from 172.39.55.119:40743 dis... jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-2 343842
19 2024-11-23T23:11:05.845-07:00 24/11/24 06:11:05 INFO BlockManager: BlockManager stopped jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-4 343842
20 2024-11-23T23:11:05.842-07:00 24/11/24 06:11:05 INFO MemoryStore: MemoryStore cleared jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-4 343842
21 2024-11-23T23:11:05.829-07:00 24/11/24 06:11:05 INFO BlockManager: BlockManager stopped jr_4647e96d21fe026036608f6e5bac17b7c97eb50874a87f47b5057917e64cf8be-5 343842
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

DAP Architecture Analysis

The AWS well-architected framework provides a comprehensive approach to designing and implementing a robust, secure efficient system in the cloud. The six pillars are as follows:

1. Operational Excellence: This pillar focuses on running and monitoring systems to deliver business value and continually improve processes and procedures.

   Focus is on

   - Automated operations: Tools like CloudFormation, AWS config, and AWS Systems Manager for automation and consistency.
   - Monitoring and Insights: CloudWatch and AWS X-Ray will be used to gain actionable insights into system performance.
   - Iterative Improvement: Use data-driven post-incident reviews to refine operations.

2. Security: AWS emphasizes protecting data, systems and assets while improving security through automation
   - Identity and access management Use AWS IAM, multifactor authentication and least privilege principles.

3. Reliability This pillar ensures that systems recover from failures and scale dynamically to meet demand.
   - Resilience Design: Use Elastic Load Balancing, Auto Scaling, and Multi-AZ deployment for fault tolerance.
   - Monitoring Health: Use Route 53 Health Checks and AWS CloudWatch for tracking system health.
   - Backup and Restore: Utilize AWS Backup and Amazon S3 versioning for data protection.

4. Performance Efficiency: This pillar focuses on optimizing resources to maintain efficient and scalable performance as demand changes.
   - **Flexible Computing**: Use AWS Lambda for serverless computing or EC2 instances with Auto Scaling.

5. Cost Optimization: AWS helps organizations balance performance with cost efficiency.

6. Sustainability: This newer pillar focuses on reducing the environmental impacts of workloads through better design and implementation.
    - Energy Efficiency: Optimize compute resources to reduce energy consumption.
    - Sustainable Regions: Choose AWS regions with renewable energy sources.
    - Resource Utilization: Leverage virtualization and serverless to maximize usage and minimize waste.

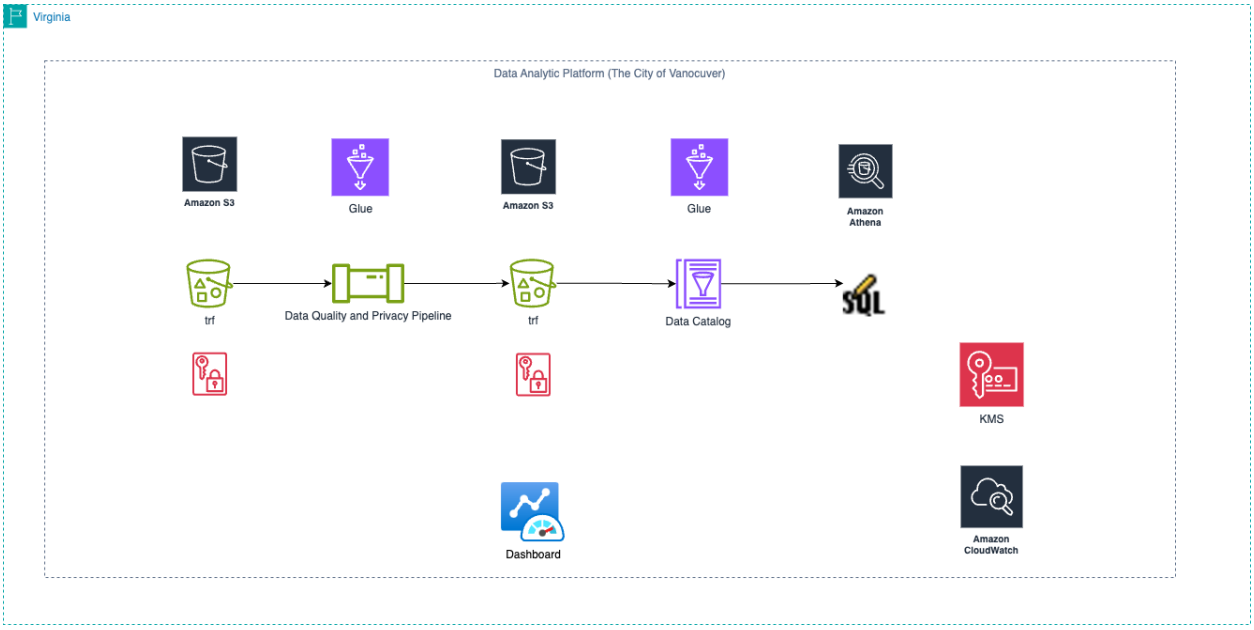
DATA ANALYTIC PLATFORM (DAP) IMPLEMENTATION FOR CITY OF VANCOUVER ISSUED OPERATING PERMITS – WATER SYSTEMS

(Team Member 3 – Samuel Ayodeji Omotayo; 2302001)

About this Project Phase and the Client:

Phase 2 of this project focuses on data protection, governance and observability for the Data Analytical Platform designed and implemented for the City of Vancouver Health Agency. The section would cover data enriching, protection, governance and observability following the framework/architecture below.

Fig 1: Architectural diagram of the Data Analytics Platform



DATA PROTECTION

Overview

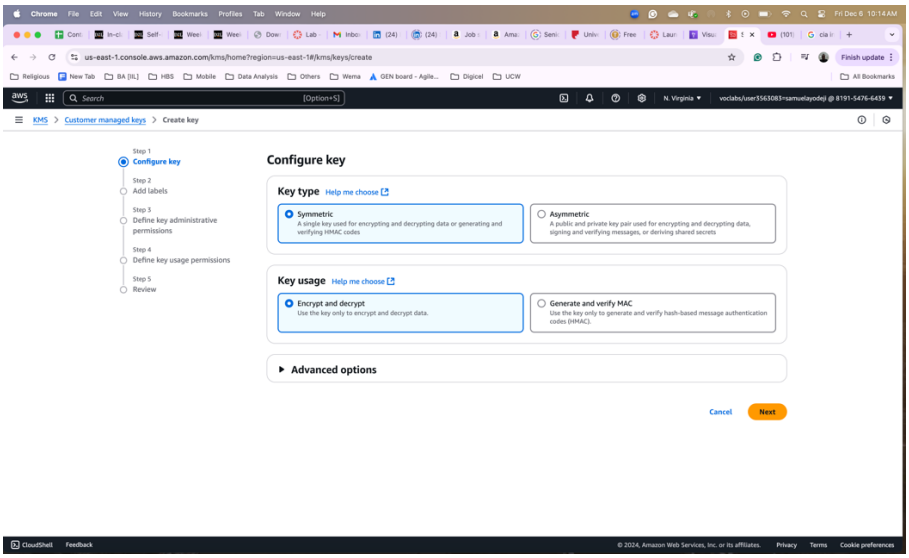
This section covers the data protection of the analytical platform designed for the City of Vancouver's City Health Agency (CHA). Data protection is critical to the well-being of any organization and in most case, data is considered the lifeline of every organization. Unprotected data could create vulnerabilities for an organization whereby unauthorized actors could gain access and tamper with the integrity (information) or even destroy these data resulting in business disruption. A cybersecurity report from IBM indicated that the estimated global average cost of a data breach is $4.8MM in 2024 (IBM, 2024). This exercise would consider the following to protect CHA's data:

- Confidentiality: Guarding access to these data by identifying who can access this data, what they are permitted to do and how they can do that.
- Integrity: Protecting against any alterations; ensuring that it is complete and authentic. Encryption and decryption would be used to protect the integrity of the data.
- Availability: Ensuring accessibility of data when needed; guarding against any disruption.

Multiple functionalities from the Amazon Web Service (AWS) Key Management System (KMS) would deployed through the process.

Task 1: Initiate a symmetric protection key

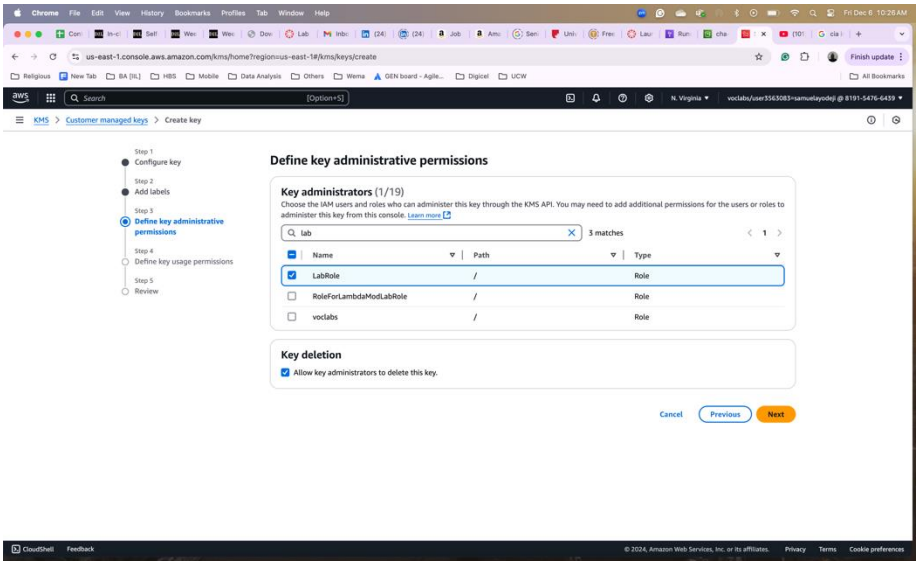
Fig 2: An AWS KMS interface to configure a symmetric key type



Note. A Symmetric Key Type enables a 2-way operation for encryption and decryption. This ensures that the inflow and outflow of data are protected. Own work.

Task 2: Define Key Administrative Permissions

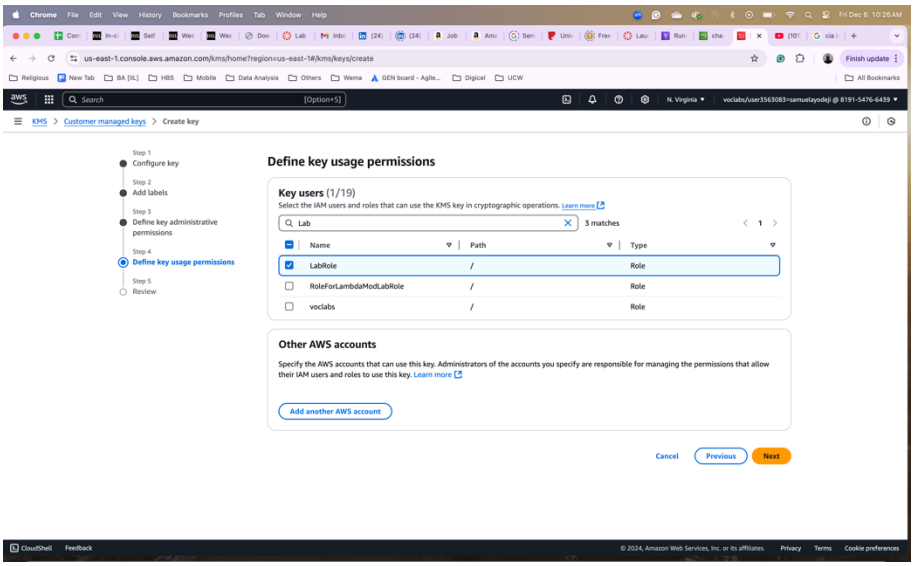
Fig 3: Assigning an Administrator to the generated Key



Note. The 'LabRole' is a role-based privilege granted to the Administrator who has access to a collection of permissions to create, delete and rotate security keys. Own work.

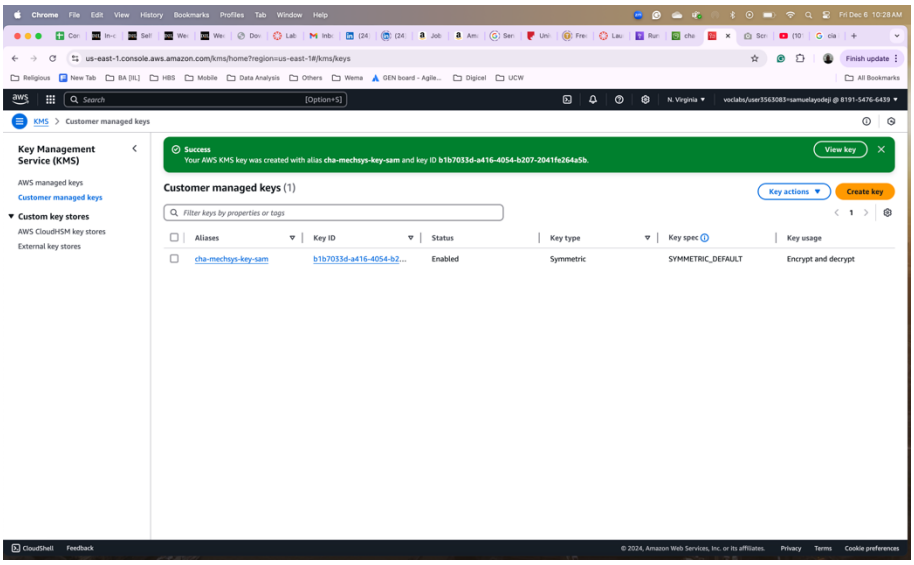
Task 3: Defining Key Usage Permissions

Fig 4: Assigning User(s) to the generated Key



Note. The users are here are employees who have been permitted to use the generated security keys or otherwise as stated by the administrator. Own work.

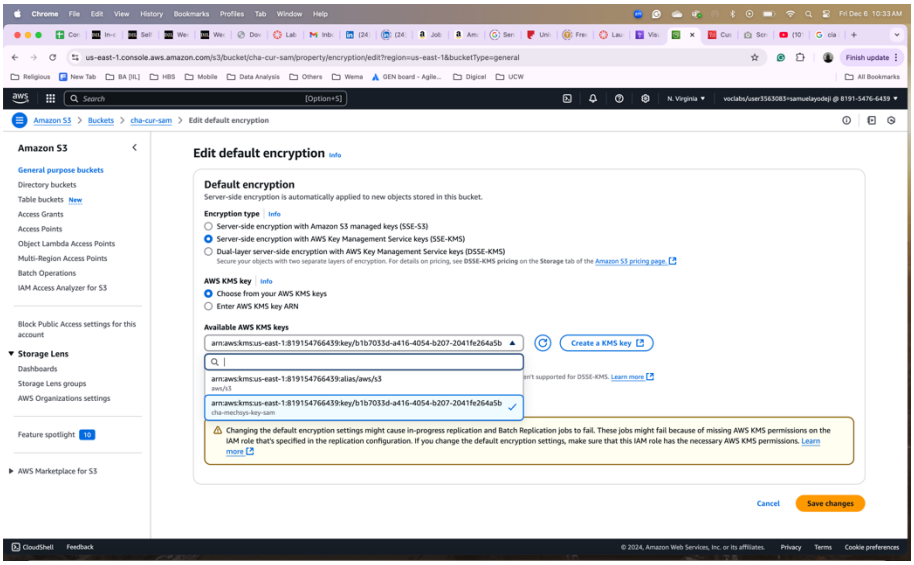
Fig 5: An AWS KMS key has been generated



Note. A security key is successfully generated to protect data inflow (ingestion) and outflow for all the buckets it would be applied. Own work.

Task 4: Applying Keys (Encryption) to all Zones

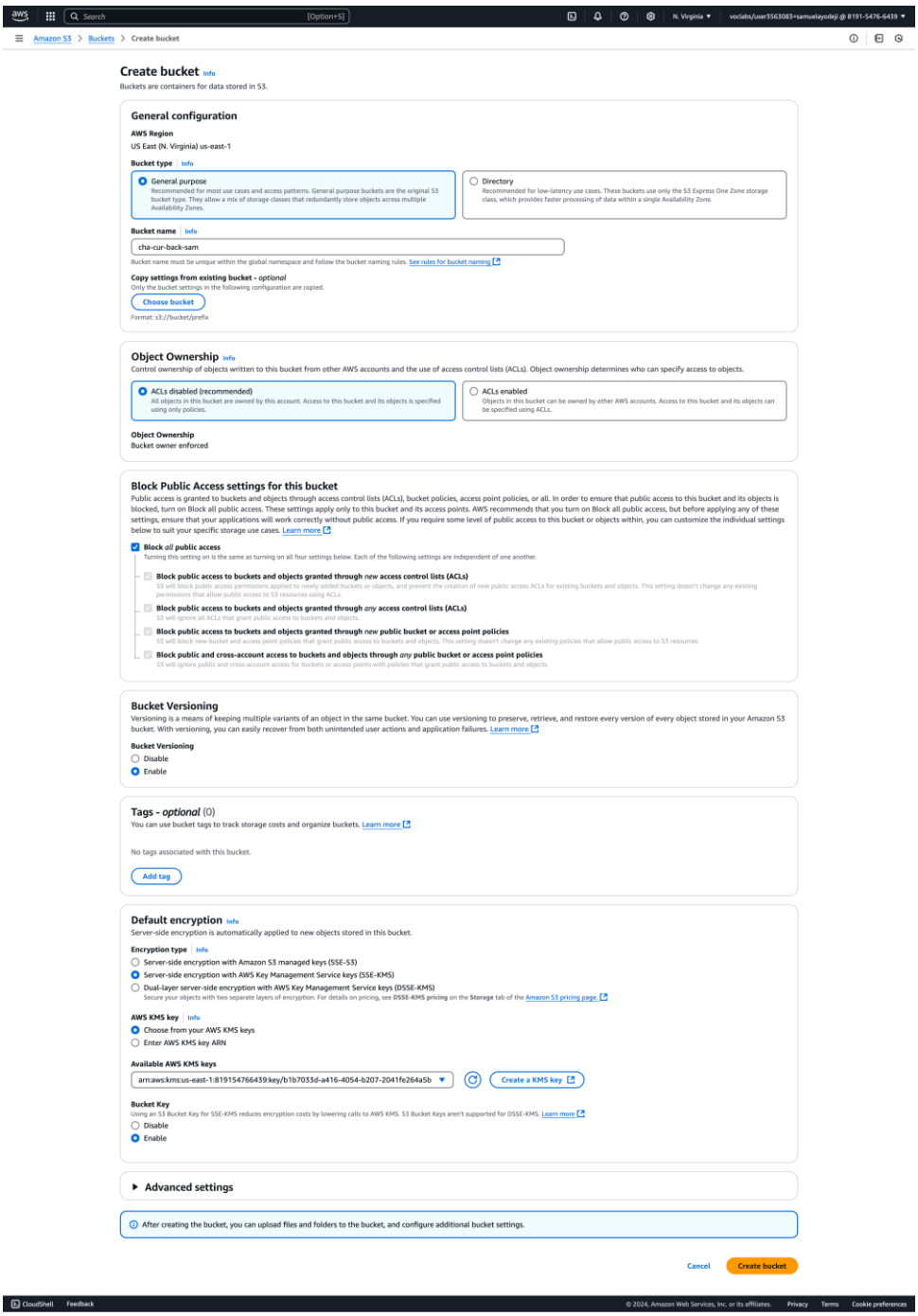
Fig 6: Applying the City Health Agency's generated encryption to a bucket.



Note. An encryption key is applied to one of the buckets. This ensures maintaining and guarding the integrity of the items stored in this bucket. Own work.

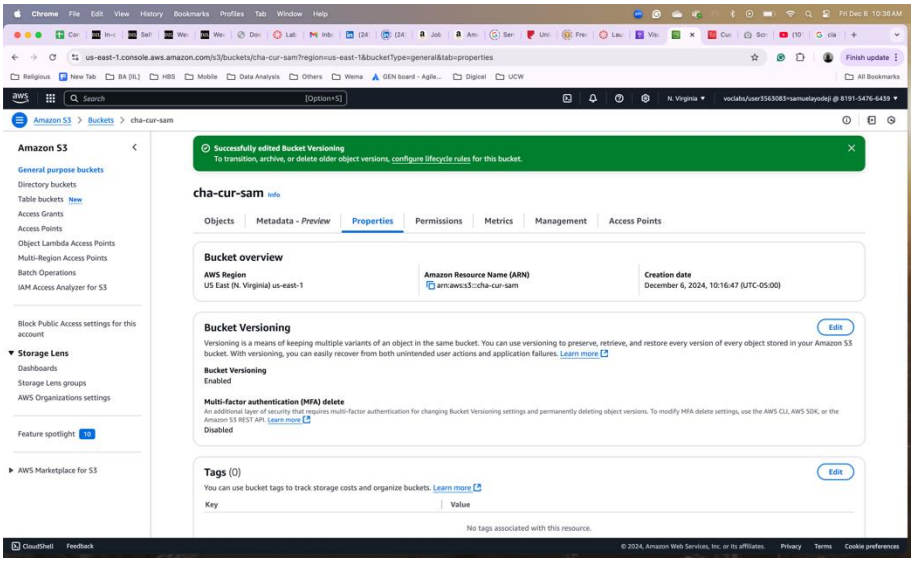
Task 5: Applying Bucket Versioning to a bucket

Fig 7: This screen showcases the process of applying Bucket Versioning to a bucket.



Note. Bucket Versions is another integrity protection measure. It automatically activates and provides a new version for a variant of any object in the bucket it is applied. Own work.

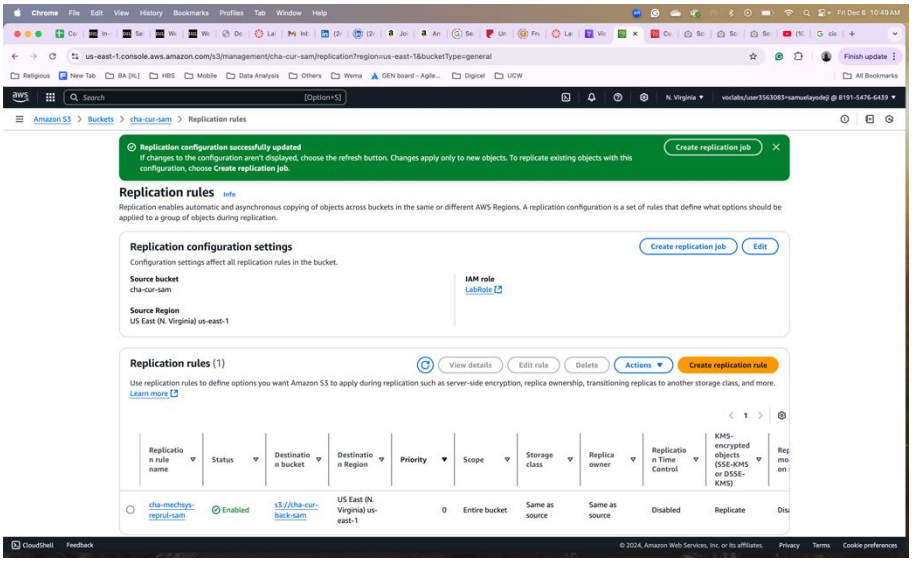
Fig 8: A successful activation of Bucket Versioning to a bucket.



Note. Successfully enabled Bucket Versioning to a curated bucket to ensure data protection measures. In this case, any alteration or update to an object would get a version of the previous state. Own work.

Task 6: Applying the Replication Rule

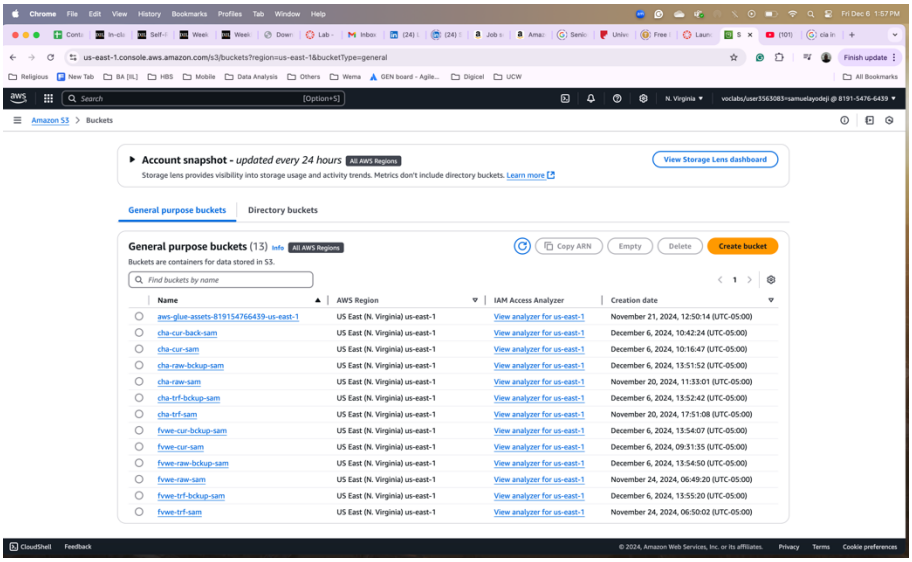
Fig 9: Replication rule activated for backup



Note. The Replication rule securely creates a backup for any object in a bucket it is applied. This is another measure to guard integrity and accessibility. It automatically creates a copy of the objects in another designated bucket. Own work.

Task 7: Protect all the zones (buckets) in CHA's Account

Fig 10: Replication rule activated for backup



Note. Created a backup bucket for all buckets with enabled protection measures, such as Bucket Versioning and Replication rule while applying the security key across board.

DATA GOVERNANCE

Overview

Data governance is the collection of processes, policies, roles, metrics, and standards that ensure data is secured, protected, private, accurate (integrity), and usable throughout the data life cycle. This includes the certain data that are managed. For example, sensitive data that contains Personally Identifiable Information (PII) are governed by certain policies to ensure that they are only accessible to authorized users. Data governance provides the following benefits:

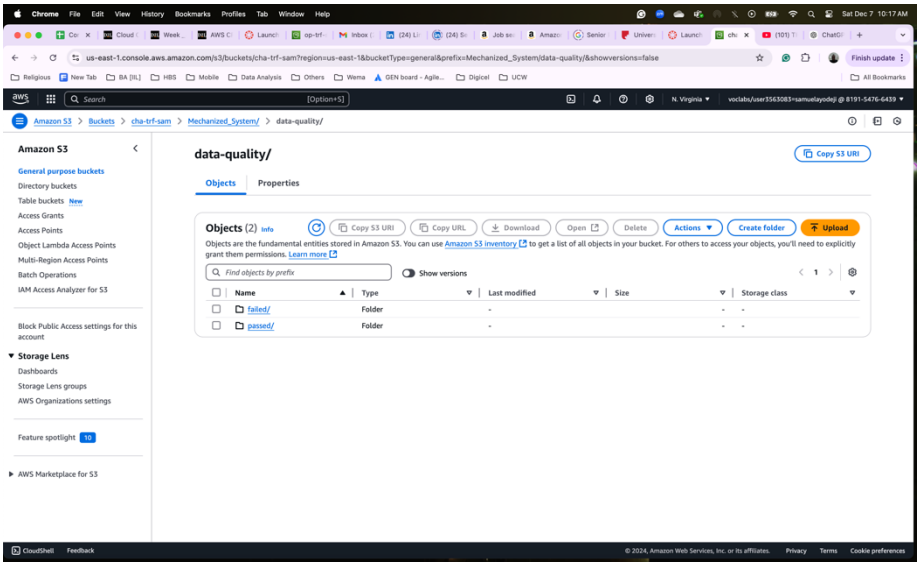
- Improves data quality
- Increases operational efficiency
- Supports regulatory compliance

The following AWS services will be used:

1. AWS S3
2. AWS Glue

Task 1: Create a folder in the transformed zone to store the outcome of the quality control assessment.

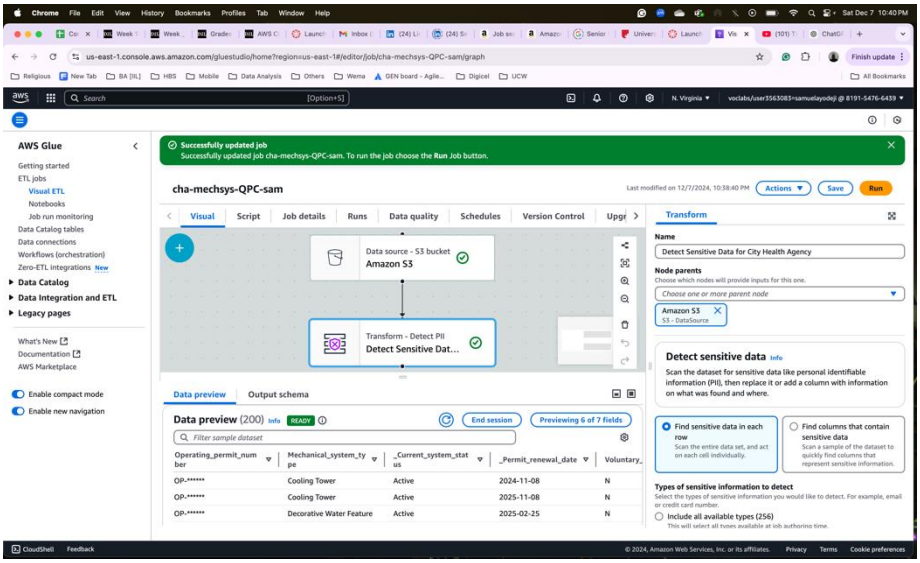
Fig 11: An AWS S3 folder 'data-quality' to store the assessment outcome



Note. A successfully created 'Data-Quality folder with two (2) sub-folders – 'failed' and 'passed' to store the outcome of each record based on the outcome. Own work.

Task 2: Create a privacy rule/check using the PII framework leverage AWS Glue

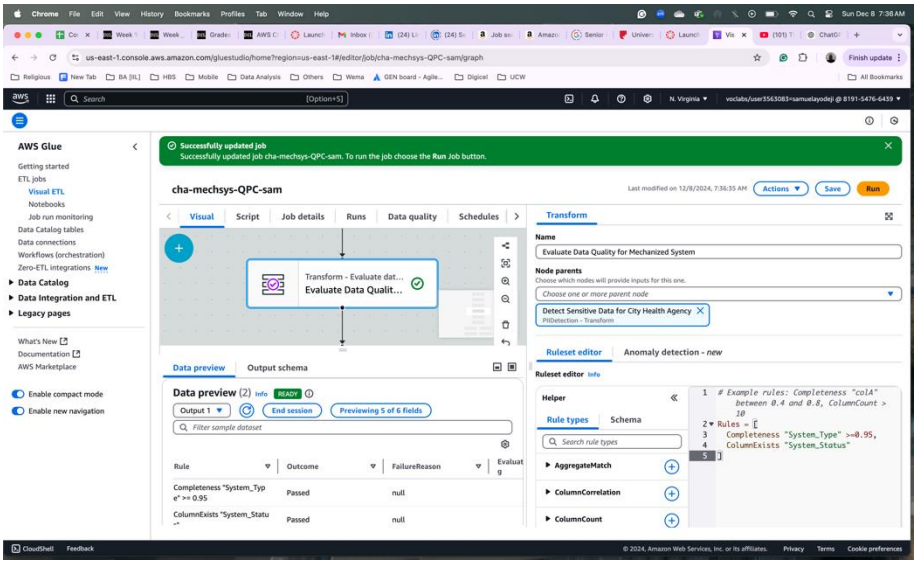
Fig 12: A data preview of redacted detected text identified as Personally Identifiable Information (PII)



Note. Using the 'Detect Sensitive Data' function in the AWS Glue to protect (privacy control) the data by replacing or masking sensitive information with special characters.

Task 3: Initiate Quality Control policy

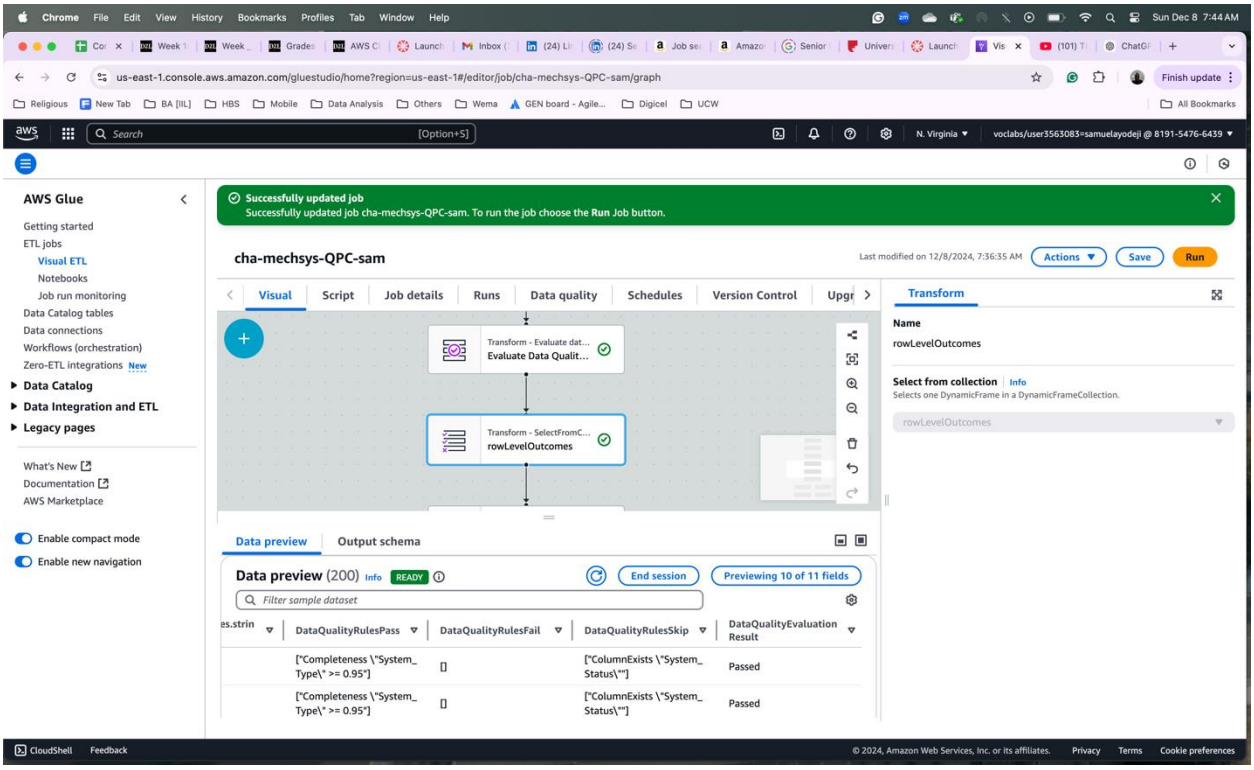
Fig 13: A preview of control policies to ensure the quality of data in our records.



Note. The 'Evaluate Data Quality' enables the use of various policies to assess the quality of the records and also provides the outcome of the assessment – either pass or fail. This ensures that all established policies are met by any records introduced or ingested into the system. In this case, I am checking against the following functions:

- *'Completeness': To confirm that there is no null value. This is critical to this record because without the knowing what type of water management mechanized system you are referring to, the entire dataset is useless.*
- *'ColumnExists': To ensure that at all times, the column record must check that the column/record that indicates if the mechanized system is Active or not is always present in the dataset.*

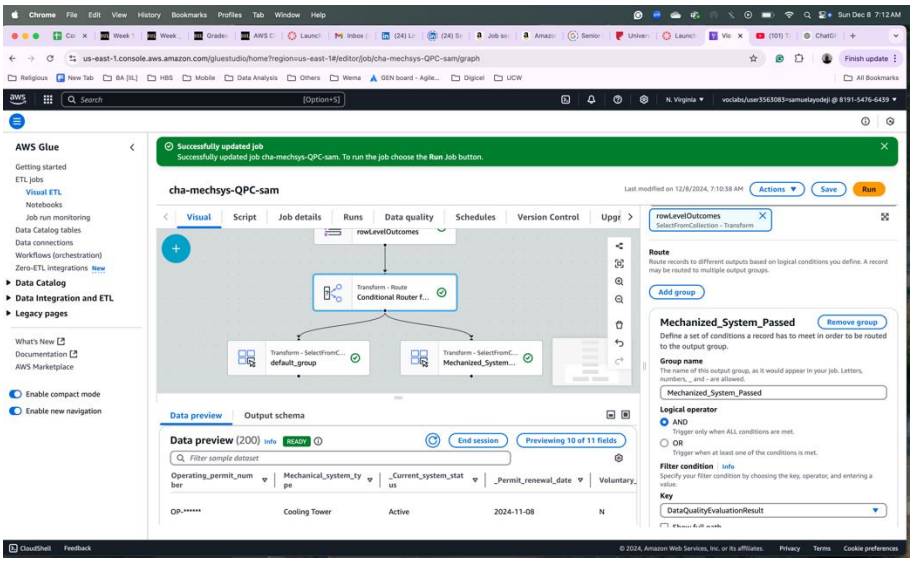
Fig 14: A preview of the implemented quality control policy.



Note. This dataset passed the two quality control measures or policies introduced.

Task 4: Aggregate the outcome of the Quality assessment

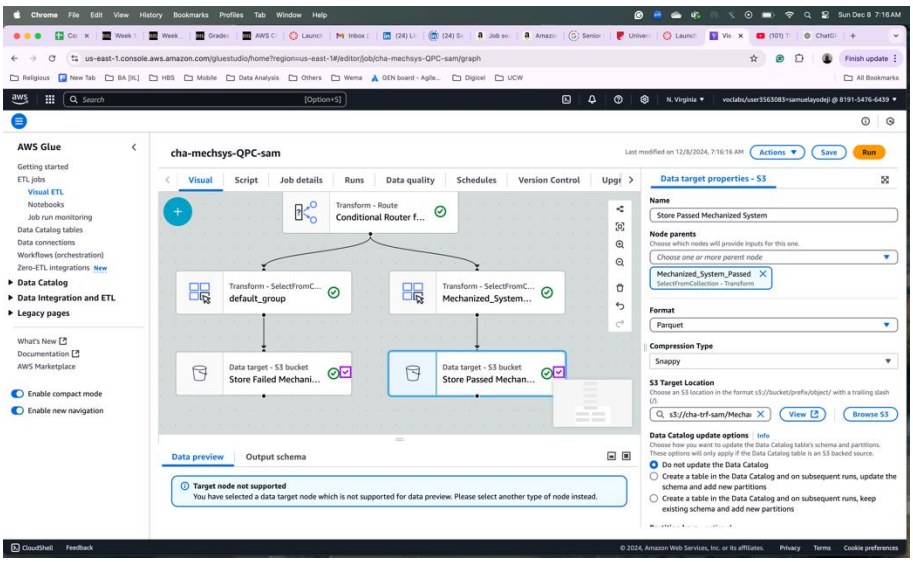
Fig 15: Introducing the 'Conditional Router' function to collate respective outcomes



Note. A conditional Router has been applied to segment the records.

Task 5: Store results in respective 'passed' or 'failed' folders

Fig 16: Channel the results into their respective folders using the 'Data target – S3 Bucket' function



Note. Conditional Router has been applied to aggregate the passed and failed assessment records into their respective and designated folders in S3.

DATA OBSERVABILITY

Overview

IBM explains Data observability as the practice of monitoring, managing and maintaining data to ensure its quality, availability and reliability across various procedures, processes, systems and pipelines within an organization (IBM, 2024). Data observability enables optimal performance by continuously measuring based on set metrics, comparing the outcome of the metrics and taking appropriate action.

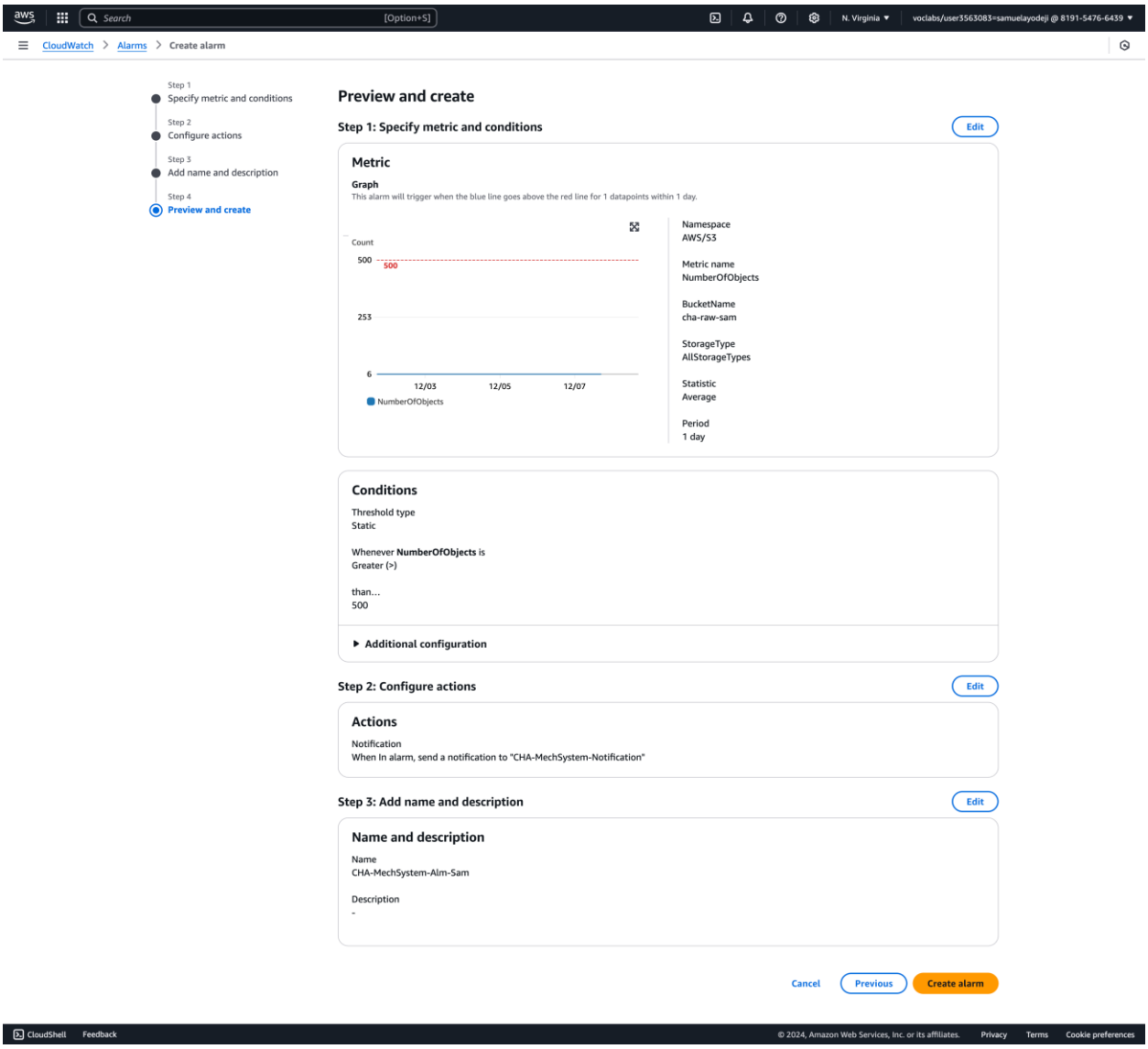
This section showcases how various metrics would be applied to the City of Vancouver Health Agency water management mechanized system Data Analytical Platform to ensure optimal performance using capabilities such as alarms and notifications, in addition to a graphical dashboard to visualize the trends across the different metrics and resources.

The following AWS services would be used:

1. AWS CloudWatch: An AWS service that helps users track, monitor, manage and control AWS resources and applications.
2. AWS CloudTrail: Used for auditing, security monitoring and operational troubleshooting by tracking users' activities through logs and account activities.

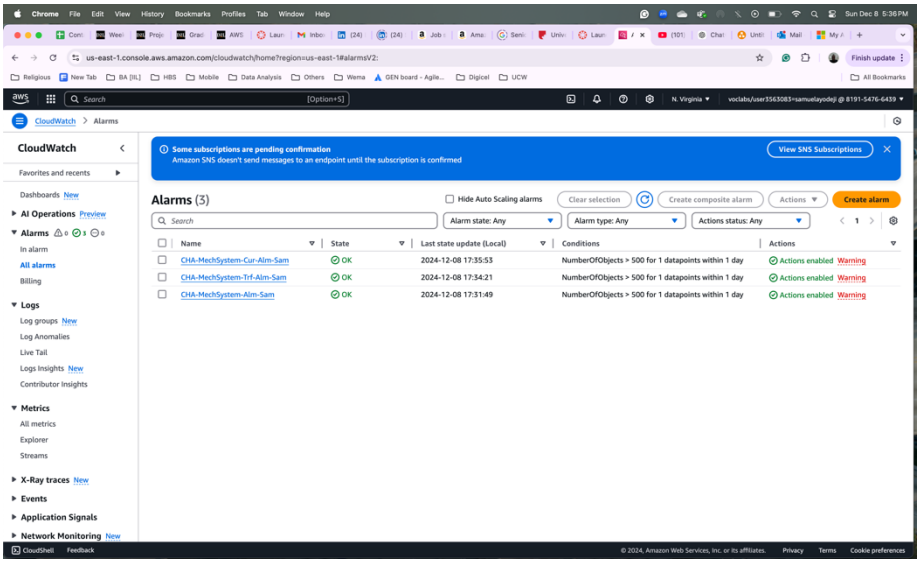
Task 1: Create the required Alarms

Fig 17: A view of the four (4) steps to creating an Alarm



Note. Introduced threshold of '500' for monitoring and control, configured an alarm system 'CHA-MechSystem-Alm-Sam' and added my school email address to receive the notification. An email alarm notification is triggered when objects exceed five hundred (500) in the raw zone within a day. A use case and instance where this is useful performance optimization where I can move other incoming objects into another folder.

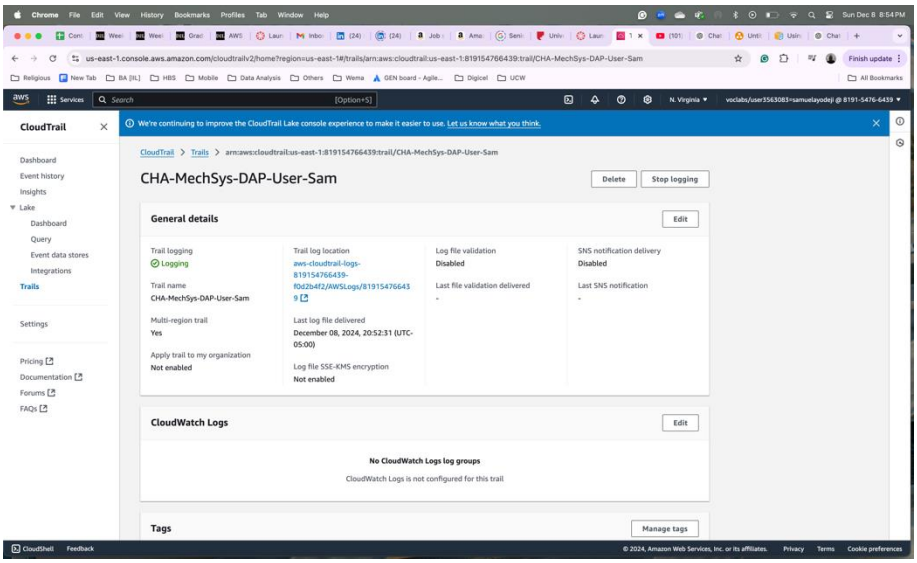
Fig 18: A list of alarms created to monitor the threshold of different resources



Note. Own work.

Task 2: Create trails to monitor user activities

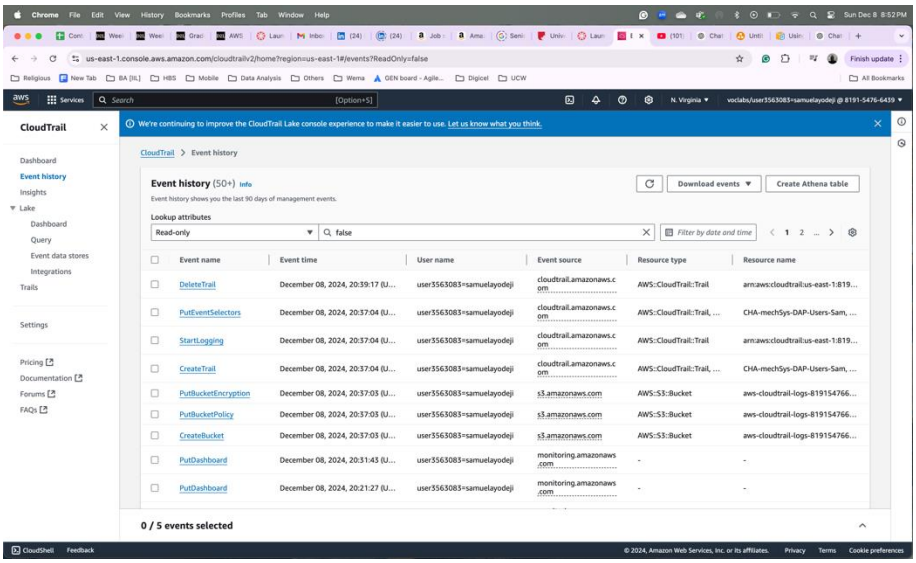
Fig 19: A trail created to monitor user activities in CloudTrail



Note. Own work.

Task 3: Review the User's Event History

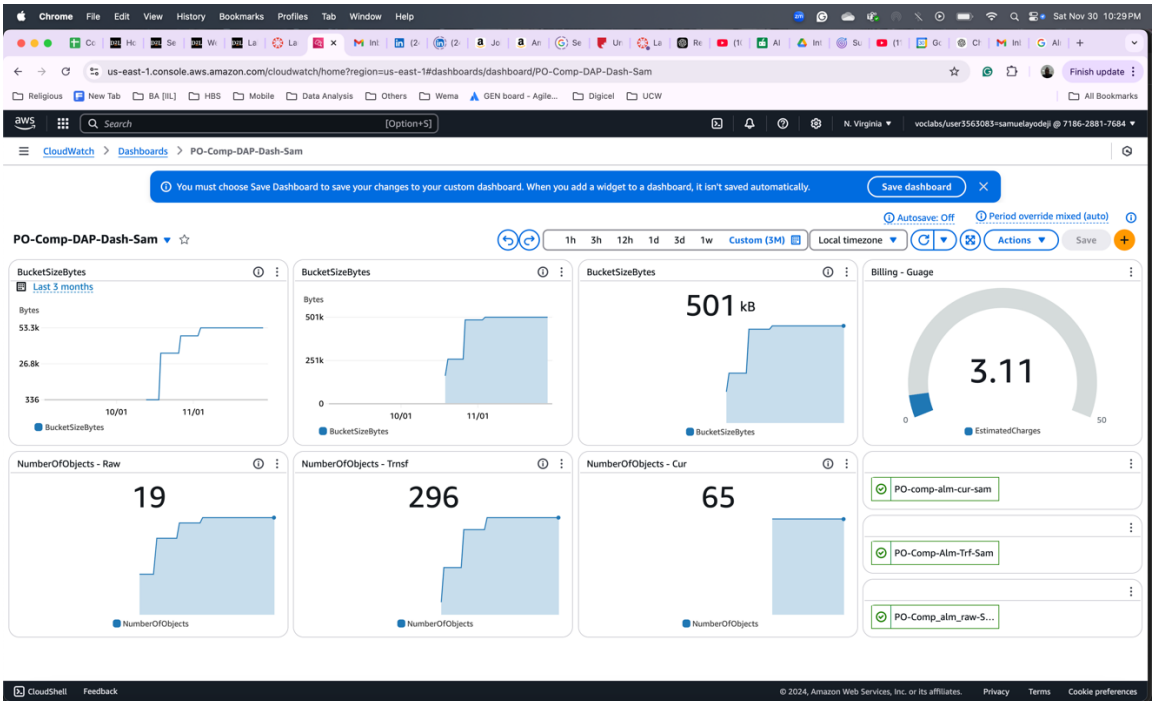
Fig 20: A view of user activities on AWS CloudTrail

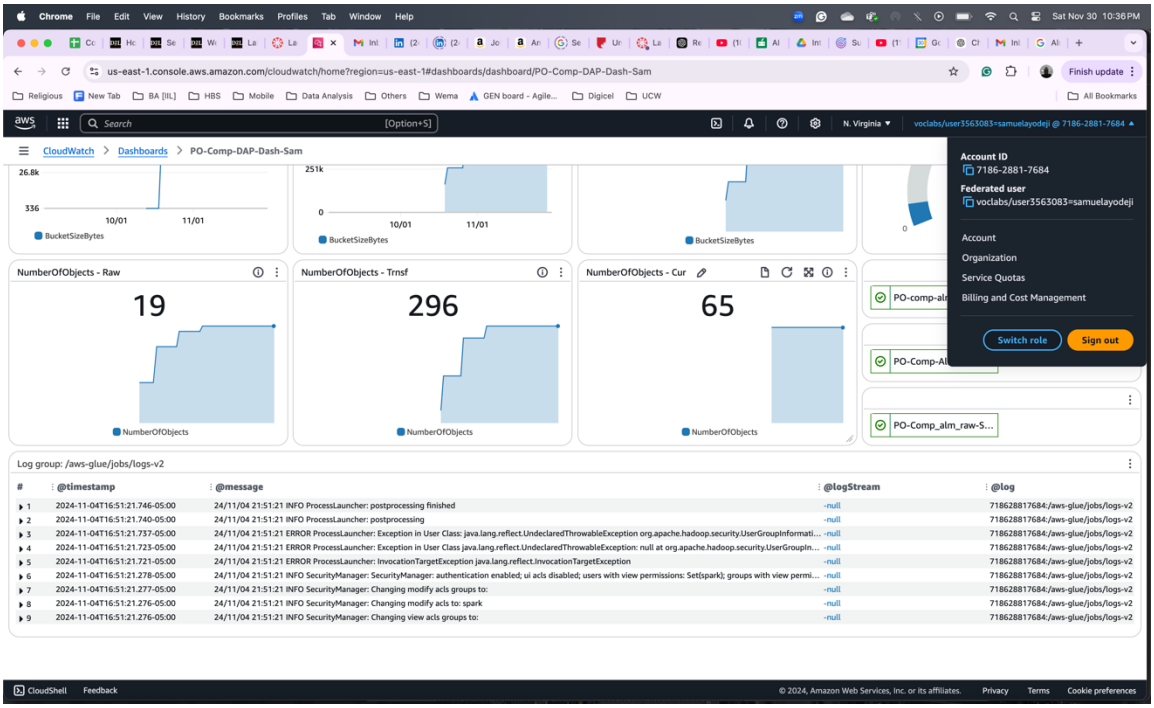


Note. A successfully created 'Data-Quality folder with two (2) sub-folders – 'failed' and 'passed' to store the outcome of each record based on the outcome. Own work.

Task 4: Review the User's Event History

Fig 21: A view of the dashboard with the different metrics





Note. A dashboard showing how the different services have are performing using – Widgets, Logs and Alarms. Own work.

DATA ENRICHING

Overview

Data Enrichment is the process of augmenting or supplementing existing data with related additional data sets from different sources. It helps improve data accuracy and completeness and crystalizes insights from data thereby helping to make better decisions.

Type of Data set: Structured data set

Type of Data source: S3 Bucket

Data Generator: File - .CSV and Excel Files from humans

Model: Relational

Output: Data Catalogue

The following AWS services would be used:

1. AWS Glue (Crawler): Crawler is found in AWS Glue. It is used to discover, prepare, move, and integrate (crawl) data from multiple sources into a single run.
2. AWS DynamoDB: Used for auditing, security monitoring and operational troubleshooting by tracking users' activities through logs and account activities.

Steps to Implementing Data Enrichment:

Task 1: Create a Table in the DynamoDB to ingest semi-structured data. Identify the primary key and the permission key.

Task 2: Still in DynamoDB, explore the (semi-data) item to view the data structure (columns) and content (rows), which will be stored in .JSON file format.

Task 3: Indicate which folder where the explored data would exported using the S3 folder link.

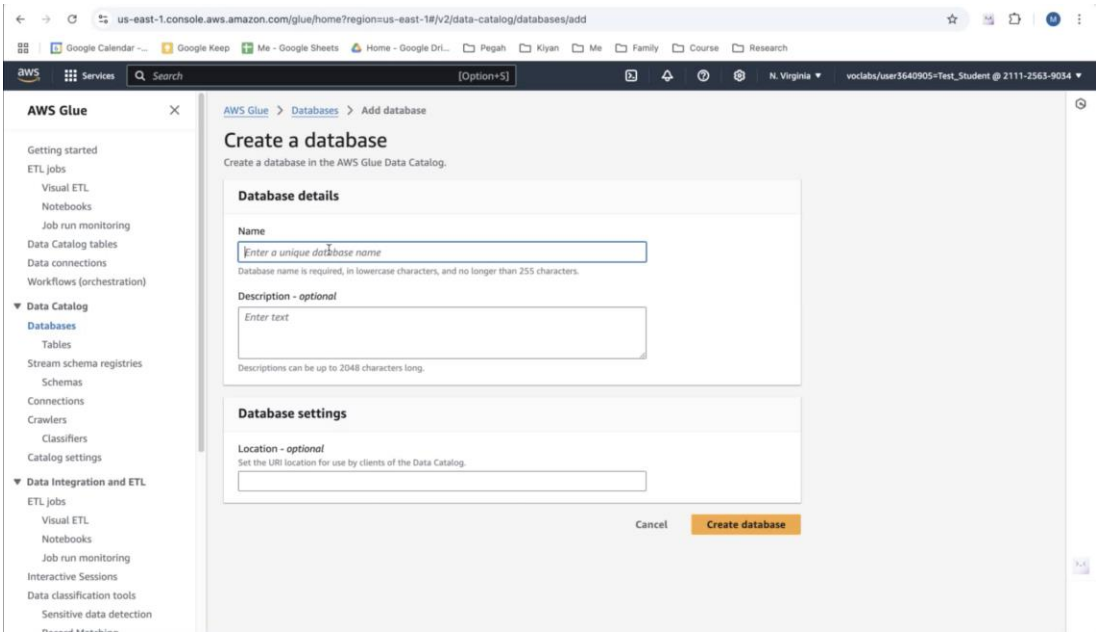
Task 4: Previous the exported or stored data from DynamoDB in S3. The DynamoDB would automatically create a different folder in the specified location.

Task 5: Create a Data catalogue – this is the collection of all the tables, the existing and newly added documents. This can be done automatically in AWS using the Crawler functionality in AWS Glue.

Steps:

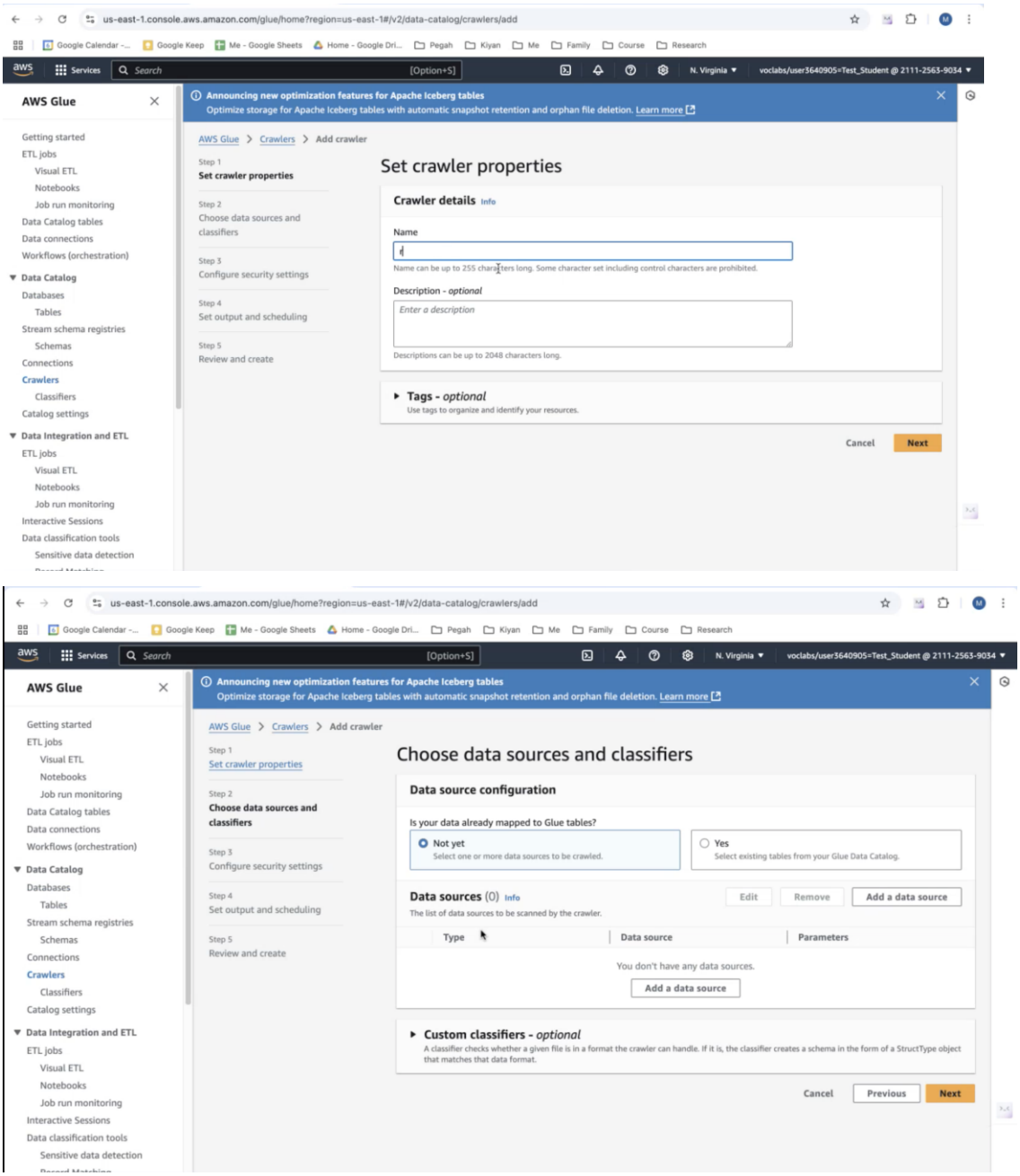
1. Create a Database (sub-item under the Data Catalogue in Glue)

   Fig 22: A interface to create a database on AWS Glue



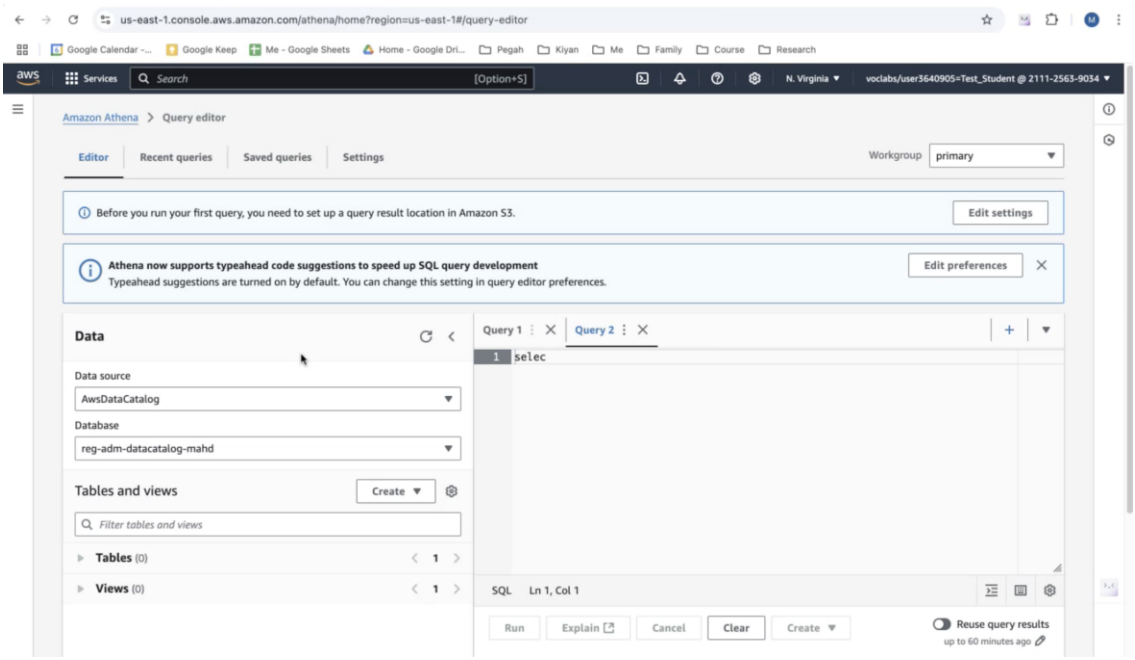
2. Add Tables using the Crawler to spool all the datasets and convert them into tables.

   Fig 23: Setting up crawler's properties for one of the identified zones.



3. View Crawler's result in the table section
4. To run (descriptive) analysis, using AWS Athena be used

Fig 24: A preview of Athena's interface that enables records query using SQL.


us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#/query-editor
Google Calendar -... Google Keep Me - Google Sheets Home - Google Dri... Pegah Kiyan Me Family Course Research
aws Services Search [Option+S] N. Virginia voclabs/user3640905=Test_Student @ 2111-2563-9034
Amazon Athena > Query editor
Editor Recent queries Saved queries Settings
Workgroup primary
Before you run your first query, you need to set up a query result location in Amazon S3.
Edit settings
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
Database
reg-adm-datacatalog-mahd
Tables and views
Create
Filter tables and views
Tables (0)
Views (0)
Query 1 Query 2
1 selec
SQL Ln 1, Col 1
Run Explain Cancel Clear Create
Reuse query results
up to 60 minutes ago

References

IBM. (2024). Cost of a data breach 2024. https://www.ibm.com/reports/data-breach

IBM. (2024, August 13). What is Data Observability? *IBM*. https://www.ibm.com/topics/data-observability