## Step 1: Data Analytical Question Formulation

A diagram of a parking lot

Description automatically generated

Goal:

Analyze the distribution of parking violations in Vancouver to identify which bylaws are most frequently violated and how the frequency of these violations changes year over year.

Descriptive Metric:

Average Number of Violations Per Bylaw Per Year:

* Description: Calculate the average number of parking violations per year associated with each specific bylaw over the past five years. This will help identify trends and highlight which bylaws are most frequently violated.
* Operational Datasets: Parking Ticket Data, including fields for violation date and bylaw number.

Diagnostic Metric:

Total Violation by Bylaw:

* Description: Determine the percentage of total parking violations that each bylaw represents. This metric will help prioritize enforcement efforts and policy adjustments.
* Operational Datasets: Parking Ticket Data categorized by bylaw number.

Predictive Metric:

Projected Trend in Violations Per Bylaw:

* Description: Based on historical data, predict whether violations for each bylaw are expected to increase, decrease, or remain stable in the coming years.
* Operational Datasets: Historical Parking Ticket Data for trend analysis.

Prescriptive Metric:

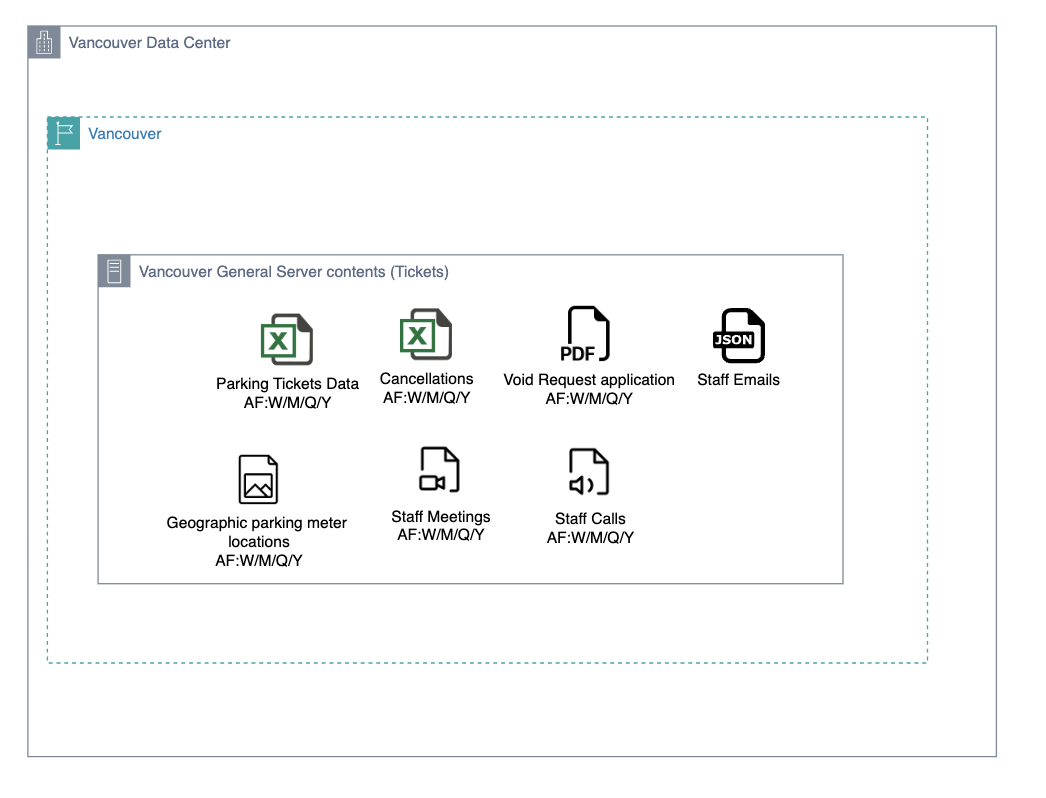
Enforcement Strategy Index:

* Description: Use the analysis of bylaw violations to inform enforcement strategies, ensuring that resources are allocated to the most problematic areas.
* Operational Datasets: Parking Ticket Data by bylaw.

Analytical Question:

"Which parking bylaws in Vancouver are most frequently violated, and how has the average number of violations per bylaw changed over the past five years?"

## Step 2: Data Discovery



In the Data Discovery phase, the focus is on gaining a comprehensive understanding of the dataset that will be analyzed. This involves finding useful information in the operational environment, in this case we can find in the general server content (Tickets) information about parking tickets, cancellations, void request, geographic parking meter locations and staff data.

## Step 3: Data Storage Design

A diagram of a computer network

Description automatically generated with medium confidence

In the Data Storage Design phase, the focus is on determining how the data will be stored in a way that optimizes accessibility, security, and performance for subsequent analysis. For this project, the data storage strategy was planned using Amazon S3, which is a scalable storage service. The design includes organizing the data by year within the Landing/ folder, which allows for easy access and management of the parking ticket data over time.

The design also considers the need for processed and raw data separation. As such, folders for Processed/, Analytics/, Logs/, and Metadata/ were created within the main Parking\_Tickets directory. This structure ensures that data is stored in an organized manner, allowing for efficient data retrieval and management. It also supports future scalability, as more data can be added to the system without disrupting the existing structure.

## Step 4: Dataset Preparation

A screenshot of a computer

Description automatically generated

In the Dataset Preparation phase, the goal is to ensure that the data is properly structured and ready for upload into the designated storage areas. This involves verifying that the raw data files, often provided in Excel format, are correctly formatted with the appropriate columns and rows. Specifically, each Excel file corresponding to a particular year should contain all the necessary information for that year’s parking tickets, including details like violation date, location, bylaw number, and any other relevant data fields.

For this project, the dataset under review consists of parking tickets issued in Vancouver, with data provided through the Vancouver Open Data Portal. The exploration of this dataset has revealed several important columns that are crucial for analysis:

* **Block**: The specific block number where the violation occurred.
* **Street**: The name of the street where the parking violation took place.
* **EntryDate**: The date when the violation was recorded.
* **Bylaw**: The bylaw number that corresponds to the specific violation.
* **Section**: The section of the bylaw that was violated.

These columns provide the fundamental information necessary for analyzing parking violations, including identifying trends over time and understanding the geographic distribution of violations.

Additionally, Appendix 1 of this report provides detailed descriptions of the specific bylaws referenced in the dataset. Each bylaw number corresponds to a particular type of violation, such as parking in a restricted area, overstaying a time limit, or blocking a fire hydrant. This appendix is a crucial reference for interpreting the data, as it links each numeric bylaw identifier to its legal definition.

During this phase, the dataset's completeness and accuracy were also assessed. Initial analysis indicated that the data is relatively clean, with no significant issues related to missing values or inconsistencies. However, a more thorough cleaning process will be conducted in later steps to ensure the data is fully ready for analysis.

## Step 5: Data Ingestion

A screenshot of a computer

Description automatically generated

In Step 5, the data ingestion process involved uploading the prepared data files to the designated storage areas within the Amazon S3 bucket. Specifically, each year’s data was stored in its respective folder under the Landing/ directory. For instance, the 2020/ folder contains the file Parking Tickets 2020.xlsx, which was successfully uploaded as shown in the attached image.

This file, sized at 9.4 MB, contains all the parking ticket data for the year 2020. The same process was followed for each subsequent year, with the corresponding Excel files uploaded into their respective folders (e.g., 2021/, 2022/, etc.). This methodical organization of the data by year ensures that the information is easily accessible and ready for further processing or analysis in the next phases of the project.

By structuring the data in this way, we not only maintain order but also facilitate efficient data management, enabling smooth transitions to future analytical tasks. The clear categorization within the S3 bucket provides a strong foundation for any data queries or analysis that will be conducted in the subsequent steps.

## Step 6: Data Storage

A screenshot of a computer

Description automatically generated

The organization of Vancouver's parking ticket data within Amazon S3 is designed for efficiency and clarity. As seen in the image, the main Parking\_Tickets folder contains several key subfolders: Analytics/, Logs/, Metadata/, and Processed/. Each of these serves a specific role in the data management process. Analytics/ stores the results of data analyses, Logs/ tracks automated processes, Metadata/ contains data dictionaries and documentation, and Processed/ holds data that has been cleaned and prepared for further analysis.

A screenshot of a computer

Description automatically generated

The Landing/ folder, shown in the second image, is organized by year (2020-2024) and serves as a staging area for new datasets. This chronological organization ensures that incoming data is stored in the correct context, making it easier to manage and retrieve when needed. Together, this structured approach facilitates secure storage, efficient data retrieval, and supports ongoing analysis and decision-making for the city's parking ticket data.

## Step 7: Data Pipeline Design

The main focus on this step is creating an efficient and streamlined process to handle the entire workflow of data preparation, processing, and publishing. The process begins with the ingestion of data into the Amazon S3 bucket (steps already finished). The "Landing" folder in S3 serves as the initial repository where raw data files from different years are aggregated. This data ingestion is crucial for centralizing all the data in a single location, making it accessible for further processing.

Once the data is stored in S3, the next step involves data cleaning and preparation using AWS Glue DataBrew. In this phase, the data undergoes a series of cleaning processes, such as handling missing values, correcting data types, and performing basic transformations. This ensures that the data is in a consistent format and ready for further processing.

After cleaning the data in AWS Glue DataBrew, the cleaned datasets are then passed to AWS Glue for more complex transformations. The data pipeline in AWS Glue is designed to handle the entire workflow, starting from schema transformations to ensure uniformity across datasets. The pipeline then filters the data to focus on relevant records, such as specific parking violations by bylaw, and aggregates this data to simplify the analysis.

The aggregated data is further refined using a SQL Query transformation in AWS Glue, where it is structured into a summary table. This table organizes the data by year and bylaw, providing a clear view of trends in parking violations over time.

Once the data has been processed and structured, it is stored back in an S3 bucket under the "Target\_TP" directory. This serves as the final data storage location, where the processed data is ready for visualization and reporting.

For data visualization, the processed data can be imported into Amazon QuickSight, AWS's business intelligence service. In QuickSight, you can create visualizations such as bar charts, line graphs, and dashboards to analyze trends in parking violations. This step is crucial for turning raw data into actionable insights.

Finally, in the data publishing step, the visualizations and reports generated in QuickSight can be shared with stakeholders via dashboards, email reports, or by embedding them in applications. The entire process, from data ingestion to publishing, ensures that data is effectively transformed, analyzed, and shared, providing valuable insights into parking violations over time.

## Step 8: Data Cleaning

A screenshot of a computer

Description automatically generated

In the image, AWS Glue DataBrew is being used to examine the parking ticket dataset, providing a detailed view of the data's structure and quality. The dataset includes nine columns, with key fields such as Block, Street, EntryDate, Bylaw, and Section. The data quality metrics shown indicate that 100% of the values in these columns are valid, with no invalid or missing values. This means that the dataset is clean and complete, which is crucial for accurate analysis.

## The image also highlights the distribution of values in the Bylaw column, showing the distinct and unique counts, which helps in understanding how frequently each bylaw is being violated. The tool provides an easy-to-read profile of the dataset, allowing for quick identification of any issues before proceeding with further data processing or analysis.

## By using Glue DataBrew, you can visually inspect and clean your data, ensuring that it meets the required standards for your analysis tasks in Athena or other services. This step is vital in the data preparation process, as it helps to guarantee the accuracy and reliability of the insights derived from the dataset.

## Step 9: Data Structuring

## A diagram of a company structure Description automatically generated

The image represents the Data Structuring process, where the raw parking ticket data is systematically filtered, grouped, and aggregated from the bottom-up to prepare it for final analysis.

1. **Filtering Rows:**

The process begins by taking the raw parking ticket data for each year, which has already been organized in separate folders, and filtering it based on the Status column. Only tickets with a status of "IS" (indicating issued tickets) are retained, as these represent actual violations. Other statuses, which are explained in Appendix 2, such as voided or canceled tickets, are excluded from the analysis. This ensures that the dataset only includes tickets that resulted in violations.

1. **Grouping Bylaw Rows:**

After filtering, the data is grouped by Bylaw within each year. This step organizes the data so that all instances of a specific bylaw violation are consolidated. Grouping by bylaws allows for a clearer analysis of how frequently each bylaw is violated within a given year.

1. **Counting Bylaw Groups:**

The grouped data is then aggregated by counting the occurrences of each bylaw violation. This counting step quantifies the number of times each bylaw was violated, providing a clear measure of violation frequency per bylaw.

1. **Aggregating Data by Year:**

Finally, the counted bylaw violations are aggregated into a general table that includes all the years. This final aggregation combines the counts from each year into a single, comprehensive table. The table allows for a longitudinal analysis, comparing the frequency of bylaw violations across different years, which helps in identifying trends and patterns over time.

Note that this table is an example, and the data used is from 2020 until 2024, the picture representation only have 3 years to be used as an example, every raw database will have the same process.

## Step 10: Data Pipeline Implementation

A screenshot of a computer

Description automatically generated

The pipeline begins by sourcing the latest parking ticket data from the Landing folder in the S3 bucket, labeled as PT\_Landing. The pipeline starts with a schema change to ensure consistency with other datasets, followed by a filtering process where only tickets with a status of "IS" (Issued) are retained. This filter step guarantees that the analysis focuses on relevant data.

After filtering, the data is aggregated by counting the number of violations per bylaw. This aggregated data is then processed through an SQL query to format the results into a table where each bylaw represents a column, and each row corresponds to a specific year. The resulting table is then stored back in an S3 bucket as the final output.

A screenshot of a computer

Description automatically generated

It's important to note that the values shown in the example table represent only a subset of the data due to AWS Glue's limitation of processing up to 1,000 rows for preview purposes. As a result, certain columns may display zero values in the preview, which would not be the case in the full dataset.

## Step 11: Data Analysis

| **Year/Bylaw** | **2849** | **2952** | **9344** | **9978** | **12086** | **13323** |
| --- | --- | --- | --- | --- | --- | --- |
| **2024** | 96,962 | 96,111 | 16 | 202 | 229 | 1 |
| **2023** | 178,577 | 183,816 | 45 | 499 | 470 | 9 |
| **2022** | 161,809 | 192,002 | 63 | 376 | 930 | 0 |
| **2021** | 143,443 | 209,321 | 52 | 166 | 1,518 | 0 |
| **2020** | 115,878 | 183,983 | 49 | 273 | 1,239 | 0 |

Analyzing the parking bylaw violations from 2020 to 2024, we observe significant trends, especially with Bylaws 2952 and 2849. Bylaw 2952 consistently has the highest number of infractions, with a peak of 192,002 in 2022 and a subsequent decline to 96,111 in the first six months of 2024. Projecting these figures to the end of the year, it's likely that the total violations for 2024 will approach 192,222, maintaining a steady level from previous years. Similarly, Bylaw 2849, which governs street and traffic regulations, recorded 161,809 violations in 2022, increasing to 178,577 in 2023. The first half of 2024 shows 96,962 violations, suggesting a possible total of around 193,924 for the year, indicating a continued upward trend.

Other bylaws, such as 12086, 9978, and 9344, which cover more specific parking issues, display lower violation numbers. However, projecting the current 2024 figures suggests these could double by year-end, bringing them closer to their historical averages. For example, Bylaw 12086 might reach around 458 violations, consistent with its trend. Bylaw 13323, related to enforcement and penalties, consistently shows minimal infractions, reflecting effective enforcement and compliance strategies.

## Step 12: Data Visualization

A screenshot of a graph

Description automatically generated

Since the students are unable to use Amazon QuickSight, the visualization for step 12 was created using a PDF exported from Excel, which contains data that was processed through the AWS Glue pipeline. The image shows a summary of parking tickets issued under different bylaws across the years 2020 to 2024.

The graph clearly illustrates trends for each bylaw over the years, with 2024 only reflecting data up to mid-year. The predicted full-year totals for 2024, shown in the analysis, suggest that the trends observed in previous years are likely to continue. This visualization effectively communicates the comparative frequency of different bylaw violations and highlights changes over time, allowing for easy identification of which bylaws are most frequently violated. This PDF visualization serves as an alternative means to display data when more sophisticated tools like Amazon QuickSight are not available, demonstrating the flexibility and adaptability required in student projects.

## Step 13: Data Publishing

A screenshot of a computer

Description automatically generated

For the final step, Data Publishing, two instances were created in AWS EC2. The first instance, named "Tickets\_ParkTick\_GS\_Nicolas," was established as a general server. This general server acts as a remote computer that users can log into and interact with as if they were sitting at a physical workstation. It's configured to handle various administrative tasks, such as managing files and running basic commands or scripts that are essential for maintaining the system's overall operation.

The second instance, "Tickets\_ParkTick\_WS\_Nicolas," serves as a web server. This instance is specifically set up to host and serve the generated PDF file, which includes the visualizations created in Step 12. The web server makes this file accessible over the internet, allowing authorized users to view or download the file through a web browser.

A screenshot of a computer

Description automatically generated

The accompanying images illustrate these instances' current states within the EC2 dashboard. Both instances are shown as running, with all checks passed, confirming they are ready and operational. The successful deployment of these EC2 instances ensures that the parking ticket data analysis results are effectively published and accessible to stakeholders, completing the entire data pipeline from question formulation to final publication.

**Parking Tickets – Step 15**

To protect the data, we use two AWS tools: AWS IAM and AWS KMS. Both will safeguard the data in the analytical environment. The operational environment is separate and should be secured during its own process.

**AWS IAM**

Figure XX - IAM Dashboard in AWS

A screenshot of a computer

Description automatically generated

*Note: Screenshot of the IAM dashboard in AWS used for identity and access management. The data is sourced from Vancouver's open data portal (*[*https://opendata.vancouver.ca/pages/home/*](https://opendata.vancouver.ca/pages/home/)*).*

With this tool, we can create users and user groups, each assigned different roles. Every role has permissions for certain applications to view or manage the data. In this project, we are not directly interacting with AWS IAM, but following the appropriate steps, we can create at least four groups:

* Group 1 (data ingestion): Employees who work on adding historical data to the Landing folder will only have permissions to add, modify, or delete data. They will have full access to the Landing folder and write permissions in S3.
* Group 2 (data analyst): This group of users will have read access to the original data, have access to the Processed and Curated folders, and the option to use the data to create derived datasets without modifying the original data.
* Group 3 (Supervisors): The supervisor group should have read access to the Analytics folders and access to the generated reports, but they will not be able to modify the data or analyses already performed.
* Group 4 (Security Auditors): Maintaining data security is vital, so a group of users must audit the behavior of previous users. This group will have access to CloudTrail and will be able to review all security logs from CloudWatch, including reviewing which users intervened and what changes were made, but they will not have access to the data or analyses.

**AWS Key Management Service (KMS)**

Figure XX - Key Management Service (KMS) in AWS

A screenshot of a computer

Description automatically generated

*Note: Screenshot of the Customer Managed Keys section in AWS Key Management Service (KMS) showing three keys used for data encryption. The data is sourced from Vancouver's open data portal (*[*https://opendata.vancouver.ca/pages/home/*](https://opendata.vancouver.ca/pages/home/)*).*

With Amazon KMS, we can create a key to encrypt our data, preventing unauthorized or external users from viewing the actual stored information.

For the project, three keys were created:

* **“ParkingTickets\_PI\_Key\_Nicolas”**: This key is for sensitive personal data, as the collected information contains sensitive personal data (vehicle registration, driver's license number, identification number, and others). This first key is created for users who need to review this data. At this stage, greater control and supervision over who can access the data is required due to current regulations. This key would mainly be used by Group 1 created in AWS IAM.
* **“ParkingTickets\_General\_Key\_Nicolas”**: This data key was created to encrypt already processed data, after the deletion or encryption of sensitive personal data. Therefore, it is a key that will have more users with access. This will allow broader access without compromising sensitive personal data. Groups 1 and 2 created in AWS IAM will have access to this key.
* **“ParkingTickets\_Analytical\_Key\_Nicolas”**: A third key will be created for processed analytical files, i.e., it will protect the data that has already been processed and is ready to be viewed or analyzed. In this case, Groups 2 and 3 will have access to this key and will be able to view the data.

It is important to note that Group 4 created in AWS IAM will only be able to review the actions of the aforementioned users, not the actual data, so they do not need to be part of the users who have acces the mentioned keys.

These two tools protect the data by focusing on three key factors:

* **Confidentiality:** Ensures that data is only accessed by validated users through AWS IAM.
* **Integrity:** Prevents data from being modified without proper authorization.
* **Availability:** Ensures validated users have access to the data while preventing unauthorized usage.

**Parking Tickets – Step 16**

In this step, we want to ensure the data complies with specific rules:

* **Data Quality:** No invalid data, NaN, null, or duplicated entries. The desired data properties are completeness (no null values), uniqueness (no duplicated rows), freshness (no outdated values), and validity (no incorrect data).
* **Data Privacy:** Personal Identity Information (PII), such as passport numbers or driver's licenses, must remain confidential.
* **Data Compliance:** The data must adhere to external regulations, including government laws.

Figure XX - AWS Data Pipeline Workflow for Parking Tickets

A screenshot of a computer

Description automatically generated

*Note: Screenshot of an AWS data pipeline workflow showing the process for fetching, detecting, and transforming parking ticket data. The data is sourced from Vancouver's open data portal (*[*https://opendata.vancouver.ca/pages/home/*](https://opendata.vancouver.ca/pages/home/)*).*

A new ETL job was created in AWS Glue, and with this tool, we can create rules to ensure that the analyzed data complies with the three aspects mentioned above. The rules for our case are:

* **Data quality**:
  + **Completeness of data**: Ensure that critical fields (such as Block, Street, EntryDate, and Bylaw) have no null or empty values. The data will be valid if it has less than 5% null or empty values.
  + **Date validation**: Ensure that the data in the EntryDate field is within the estimated and expected date ranges, for example, between 2020 and 2024.
  + **Duplicate verification**: A rule is created to avoid duplicated observations by analyzing the unique values of each observation.
* **Data Privacy**:
  + **Encryption or deletion of personal information**: For our evaluation, personal data is not required, as we are only measuring the total number of violations by bylaw type, so the columns with sensitive personal information were deleted. However, other analyses may require the use of sensitive information, in which case personal information should be encrypted.
  + **Anonymization of sensitive data**: In our case, it was not necessary since we did not use personal data. However, for other analyses, presenting anonymized data is ideal.
* **Data Compliance**:
  + **Bylaw validation**: Ensure that each observation matches a valid and current Bylaw. A predefined list of bylaw codes was used to perform the validation.
  + **Compliance with internal rules**: Analyses may incur internal rules (for example, the removal of last names to ensure anonymity). In these cases, a new rule must be created to comply with internal regulations.

**Parking Tickets – Step 17**

The final step is to monitor and control the process using two AWS tools:

**AWS CloudTrail:** This tool helps monitor the users created in step 15 and their activities. It provides control over access and identifies the responsible individuals for each action. Unfortunately, without a real process, we can only structure the use of CloudTrail. In our case, Group 4 created in IAM will have access to CloudTrail to review the actions of users in Groups 1 and 2 (who can modify and analyze the data).

**AWS CloudWatch:** This service creates a dashboard to track various metrics.

Figure XX - AWS CloudWatch Alarm Dashboard for Parking Tickets

A screenshot of a computer

Description automatically generated

*Note: Screenshot of the AWS CloudWatch Alarm Dashboard monitoring AWS Glue, S3 resource usage, and estimated charges for parking ticket data processing. The data is sourced from Vancouver's open data portal (*[*https://opendata.vancouver.ca/pages/home/*](https://opendata.vancouver.ca/pages/home/)*).*

In CloudWatch, a dashboard was created to monitor some critical process conditions in real-time:

* **Resource usage in AWS Glue**:
  + Execution time of AWS Glue jobs.
  + Amount of data processed per Glue job.
  + Failures in Glue jobs.
* **Storage usage in AWS S3**:
  + Total space used in S3.
  + Number of PUT and GET requests in S3 (interaction with the data in S3).
* **EC2 usage for data visualization**:
  + CPU and memory used by EC2 instances.
  + Uptime of EC2 instances.
* **Security metrics with AWS CloudTrail**:
  + IAM activities (creation/modification of users and roles).
  + Unauthorized or failed access attempts to resources.
* **Estimated AWS costs**:
  + Total estimated cost.

Additionally, to control the monitored metrics, alarms can be created in case any metric exceeds the desired limit. Actions can also be applied if metrics exceed the limit. These actions can range from stopping a process to sending an email.

The alarms created in this case were:

* **Execution Time Alarm in Glue**: If the execution time of an AWS Glue job exceeds 2 hours, an error email will be sent.
* **Glue Job Failure Alarm**: If a Glue job fails more than once consecutively, a notification email will be sent to the responsible party.
* **S3 Space Usage Alarm**: If the space used in S3 exceeds 80% of the total, an email notification will be sent.
* **EC2 Usage Alarm**: If the CPU or memory usage of EC2 instances exceeds 80% of capacity, an email will be sent to the responsible party.
* **Unauthorized Access Alarm in CloudTrail**: If there are failed or unauthorized access attempts to encrypted data or IAM resources, a high-priority email will be sent, and access to the database will be restricted for all users.
* **Cost Alarm**: If AWS estimated costs exceed 50% of the initially estimated cost, an email will be sent to supervisors.

*It should be noted that alarms and monitoring widgets directly depend on the tools used. For example, in our case, the EC2 instances were deleted, so a widget or alarm cannot be created if there is no active EC2 instance.*

# Appendix 1.- Bylaws

**Bylaw 2952: Parking Meter Bylaw**

This bylaw regulates the use of parking meters across Vancouver. It includes provisions for the operation of parking meters, payment methods, and the fees associated with metered parking. For example, this bylaw was amended to include an additional fee for payments made via a mobile phone system.

**Bylaw 2849: Street and Traffic Bylaw**

This bylaw governs the use of streets, including traffic control and the regulations for parking on city streets. It covers everything from the size and weight limits for vehicles to the rules for parking in various areas of the city. It’s a comprehensive bylaw that ensures the safe and orderly use of public roads and sidewalks.

**Bylaw 12086: Amendments and Miscellaneous Parking Regulations**

This bylaw includes amendments to existing parking regulations, often focusing on specific issues such as accessible parking, parking permits, and other specialized parking needs. For example, it has been used to update the requirements for accessible parking in strata-titled properties.

**Bylaw 9978: Related to Parking Restrictions**

This bylaw often deals with restrictions on parking in specific zones or areas within the city. It may set out the rules for residential parking permits or limit parking in certain commercial or high-traffic areas.

**Bylaw 9344: Specific Area Parking Regulations**

This bylaw could be focused on particular zones or districts within Vancouver, setting out parking rules that apply specifically to those areas. It helps manage parking congestion in areas with high demand, such as downtown or near popular attractions.

**Bylaw 13323: Enforcement and Penalties**

This bylaw likely outlines the enforcement mechanisms for parking violations, including fines and penalties for non-compliance. It might also detail the procedures for disputing a parking ticket or the processes involved in escalating enforcement actions.

# Appendix 2.- Status

The status of each parking ticket provides essential information about the outcome or current state of the violation. Below are the descriptions of the various statuses that appear in the parking ticket dataset:

* **CA (Courtesy Cancellation):**  
  This status indicates that the ticket was canceled as a one-time courtesy. This status is no longer in use, as the courtesy cancellation policy has been discontinued.
* **IS (Issued):**  
  Tickets with the status "IS" represent those that were officially issued and are considered valid violations. These are the tickets included in the analysis since they reflect actual enforcement actions.
* **RA (Cancelled due to Paid by Phone):**  
  This status applies to tickets that were initially issued but later canceled because the parking fee was paid via a mobile phone application, typically after the ticket was issued but within a grace period.
* **VA (Void):**  
  Tickets with a "VA" status have been voided, meaning they were canceled and are no longer valid. This might occur due to administrative errors or other reasons for invalidating the ticket.
* **VR (Void Request):**  
  The "VR" status is used when a void request has been made, but the ticket is not yet officially voided. This status indicates that the ticket is pending cancellation.
* **VS (Auto-void):**  
  Tickets marked as "VS" have been automatically voided, possibly due to system errors, duplicate issuance, or other automated checks that identified the ticket as invalid.
* **WR (Warning):**  
  The "WR" status indicates that the ticket was issued as a warning rather than a fine. These tickets do not carry any penalties and are intended to notify the driver of a violation without imposing a monetary fine.