Architecture for Traffic Detection Application

Documentation Information

- **Document Owner: Shankar**
- **Last Update: 06-Nov-2023**
- **Document Status: Draft**

Table of Contents

Executive Summary:

Purpose of the Document : Traffic congestion is a persistent issue in many urban areas, leading to wasted time, fuel, and increased pollution. Effective traffic detection systems can help manage congestion by providing real-time data and optimizing traffic flow. Road safety is a paramount concern. Traffic detection systems can monitor traffic for accidents, erratic driving behavior, or other safety hazards, enabling quicker responses from emergency services. Efficient traffic management can reduce travel times and fuel consumption, benefiting both individuals and businesses. Traffic detection systems can help optimize signal timings and route traffic efficiently.

Overview of the AWS Solution: Building a traffic detection application solution on Amazon Web Services (AWS) involves designing and implementing an architecture that can collect, process, and analyze traffic data efficiently. Using some of the AWS services like Dynamodb, Lambda, EC2 ,EMR,S3.

Key Benefits of AWS: Scalability, Reliability and High Availability, Global Reach,Cost Efficiency ,Variety of Services, Flexibility, Machine Learning and AI, Quick Deployment.

Project Scope:

The project aims to develop a real-time traffic detection application using AWS, focusing on improving traffic safety, optimizing congestion management, and enhancing overall transportation efficiency within a defined urban area. The scope encompasses the collection and analysis of traffic data from various sources, integration with existing transportation systems, and the deployment of a user-friendly application.

Architecture Diagram:

Architecture for Traffic Detection Application
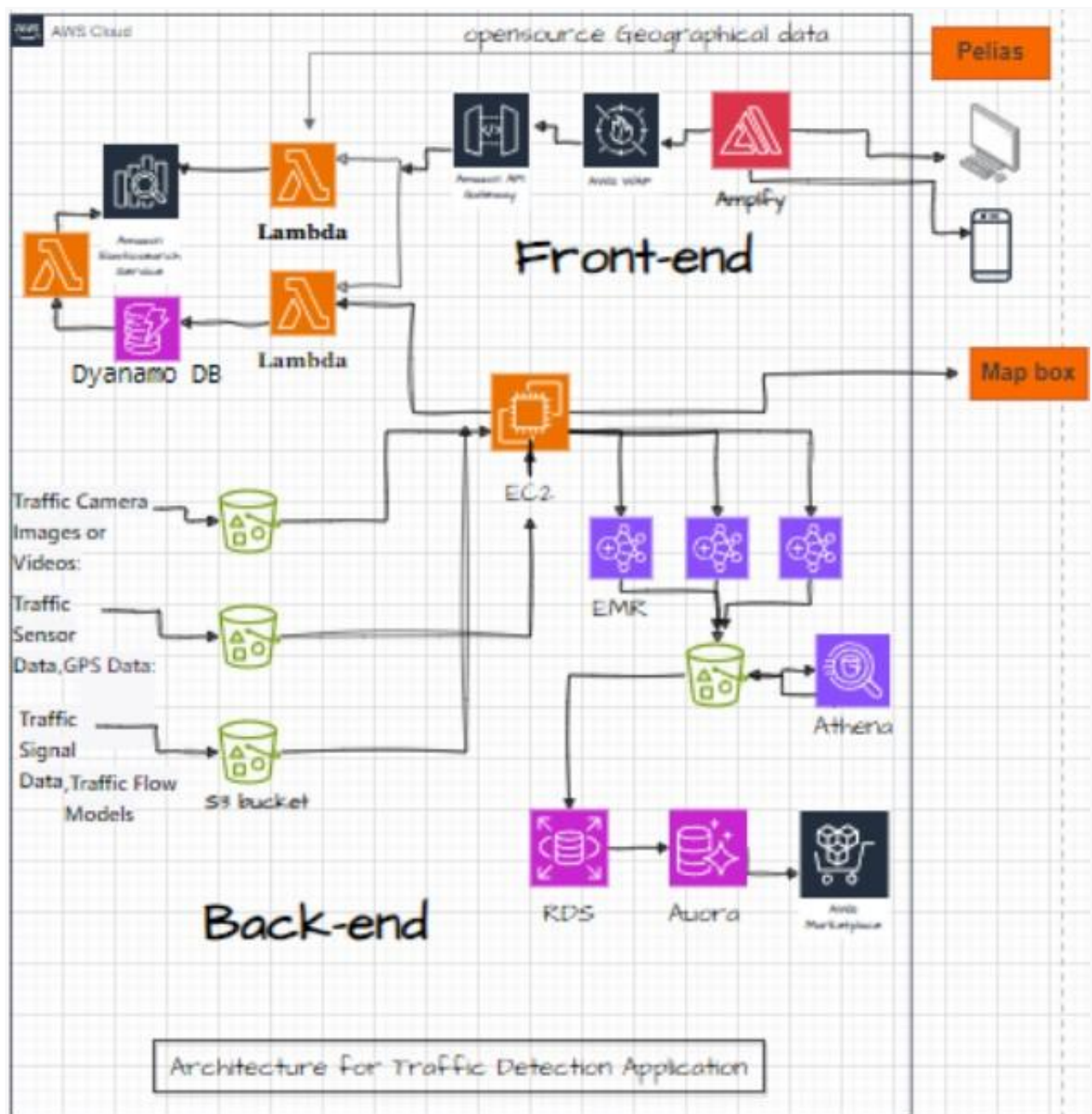
AWS Services and Technologies:

- Amazon EC2 (Elastic Compute Cloud)
- Amazon S3 (Simple Storage Service)
- Amazon Lambda
- Amazon RDS (Relational Database Services)

- DynamoDB
- AWS WAF (Web Application Firewall)
- AWS Amplify

- Amazon EMR (Amazon Elastic MapReduce)
- AWS Athena
- AWS Marketplace
- Amazon API Gateway
- Amazon ElasticSearch Service
- AWS Aurora

Justification for Service Selection:

EC2:

1. Amazon EC2 provides scalable compute resources for processing real-time traffic data, accommodating fluctuations in demand and ensuring timely data analysis.
2. EC2 instances offer flexibility to customize the application's architecture, deploy specific software components, and efficiently manage compute resources for traffic detection and analytics.

S3:

Amazon S3 is used in the traffic detection application to securely and cost-effectively store and manage large volumes of traffic data, making it easily accessible for analysis and ensuring data durability and scalability.

Lambda:

Using AWS Lambda in the traffic detection application enables cost-efficient, serverless, event-driven data processing, ensuring optimal resource allocation and scalability while minimizing operational overhead.The purpose of using Lambda between EC2 and DynamoDB is to facilitate serverless data processing, enabling real-time traffic data analysis, transformation, and seamless integration with DynamoDB for efficient and scalable data storage.

Relational Database Service(RDS):
Using Amazon RDS (Relational Database Service) for the traffic detection application provides a scalable and fully managed database solution, ensuring efficient storage and retrieval of metadata and configuration data, while offloading database management tasks and enhancing application reliability.

EMR:
Amazon EMR (Elastic MapReduce) can efficiently process and analyze large volumes of traffic data, making it ideal for a traffic detection application. Its distributed computing capabilities and compatibility with various big data tools enable timely insights and real-time traffic management, improving overall road safety and transportation efficiency.

Amplify:
Amplify simplifies and accelerates the development of the traffic detection application on AWS by providing a comprehensive set of tools and services for building and deploying scalable, secure, and responsive applications.Amplify also offers features for authentication, real-time data synchronization, and robust back-end services, reducing development time and effort while ensuring seamless user experiences.

Athena:
Athena, an AWS service, is beneficial for the traffic detection application due to its efficient query performance and cost-effectiveness in analyzing large datasets, allowing for real-time traffic insights and data-driven decision-making.

Map box:
Mapbox is utilized in this traffic detection application for its robust mapping and geospatial capabilities, enabling real-time visualization of traffic data, advanced route optimization, and geospatial analysis, enhancing the application's overall effectiveness in traffic management.

Pelias:
Pelias, a geocoding and search engine, is ideal for a traffic detection application due to its robust location data capabilities and geospatial search features, enabling precise and efficient analysis of traffic patterns, incident detection, and routing optimization.

Deployment Process:

- Utilize AWS Elastic Beanstalk or Amazon EC2 for hosting the application code and AWS RDS for database storage, configuring security groups, IAM roles, and auto-scaling as needed.
- Leverage AWS CloudFormation or Terraform for infrastructure as code (IAC) to automate resource provisioning and deployment, ensuring scalability, high availability, and efficient project management.
- To deploy the web application on AWS, use AWS Elastic Beanstalk for streamlined deployment and scaling, or AWS EC2 instances with web servers and databases for more custom control and flexibility.