

University of Bahrain

Department of Computer Science

ITCS448 – Cloud Computing

2023/2024 First Semester

Designing a Scalable and Reliable Smart Traffic Management System

Done by:

|  |  |
| --- | --- |
| Maha Mohammed | 202002565 |
| Noor Jaafar | 202007922 |
| Adnan Mohammed Redha | 202008997 |
| Aisha Jassim Altemimi | 202007058 |
| Sayed Ahmed Khalaf | 202007602 |
| Fai Omar Albabtain | 202011749 |

**Table of Contents**

[Problem Description 4](#_Toc167340049)

[Main Objectives 4](#_Toc167340050)

[Infrastructure Design 5](#_Toc167340051)

[1. Amazon Kinesis Video Streams 6](#_Toc167340053)

[2. Storing Lambda 6](#_Toc167340055)

[3. Amazon S3 6](#_Toc167340057)

[4. Retrieving Lambda 7](#_Toc167340059)

[5. Network Load Balancer 7](#_Toc167340061)

[6. Auto Scaling Group 7](#_Toc167340063)

[7. Amazon EC2 Instances 7](#_Toc167340065)

[8. Private and Public Subnets 7](#_Toc167340067)

[9. NAT Gateway 7](#_Toc167340069)

[10. Amazon DynamoDB 8](#_Toc167340071)

[11. AWS IoT Core 8](#_Toc167340073)

[Procedure: 8](#_Toc167340076)

[**1-** **VPC** 8](#_Toc167340077)

[2- **NAT** 10](#_Toc167340078)

[**3-** **Routing Table** 11](#_Toc167340079)

[**4-** **Security Groups** 11](#_Toc167340080)

[ **Load Balancer** 12](#_Toc167340081)

[ **EC2** 13](#_Toc167340082)

[**6-** **DynamoDB** 15](#_Toc167340083)

[**7-** **Target groups and load balancers** 15](#_Toc167340084)

[**9-** **AMI** 18](#_Toc167340085)

[**10-** **Auto-scaling** 18](#_Toc167340086)

[Cost Estimation 19](#_Toc167340090)

[Conclusion 23](#_Toc167340091)

# Problem Description

Traffic congestion is a significant issue in urban areas, leading to increased travel times, higher fuel consumption, and elevated pollution levels. Efficient traffic management is essential for improving urban mobility and reducing environmental impact. The Smart Traffic Management (STM) system aims to optimize traffic flow by dynamically adjusting traffic signal timings based on real-time data collected from various sources such as cameras and sensors. This system leverages AWS services to handle image processing, data processing, and storage to enhance traffic management efficiency.

# Main Objectives

1. **Optimize Traffic Flow:** Reduce congestion and minimize waiting times at intersections.
2. **Data Collection and Processing:** Efficiently collect and process real-time traffic data.
3. **Dynamic Traffic Light Control:** Adjust traffic signal timings based on real-time traffic conditions.
4. **Scalability and Reliability:** Ensure the system can handle increasing data volumes and maintain high availability.
5. **Security and Privacy:** Protect sensitive data during transmission and storage.

# Infrastructure Design

The STM system is designed using AWS services to create a robust, scalable, and secure architecture. The key components of the infrastructure include:

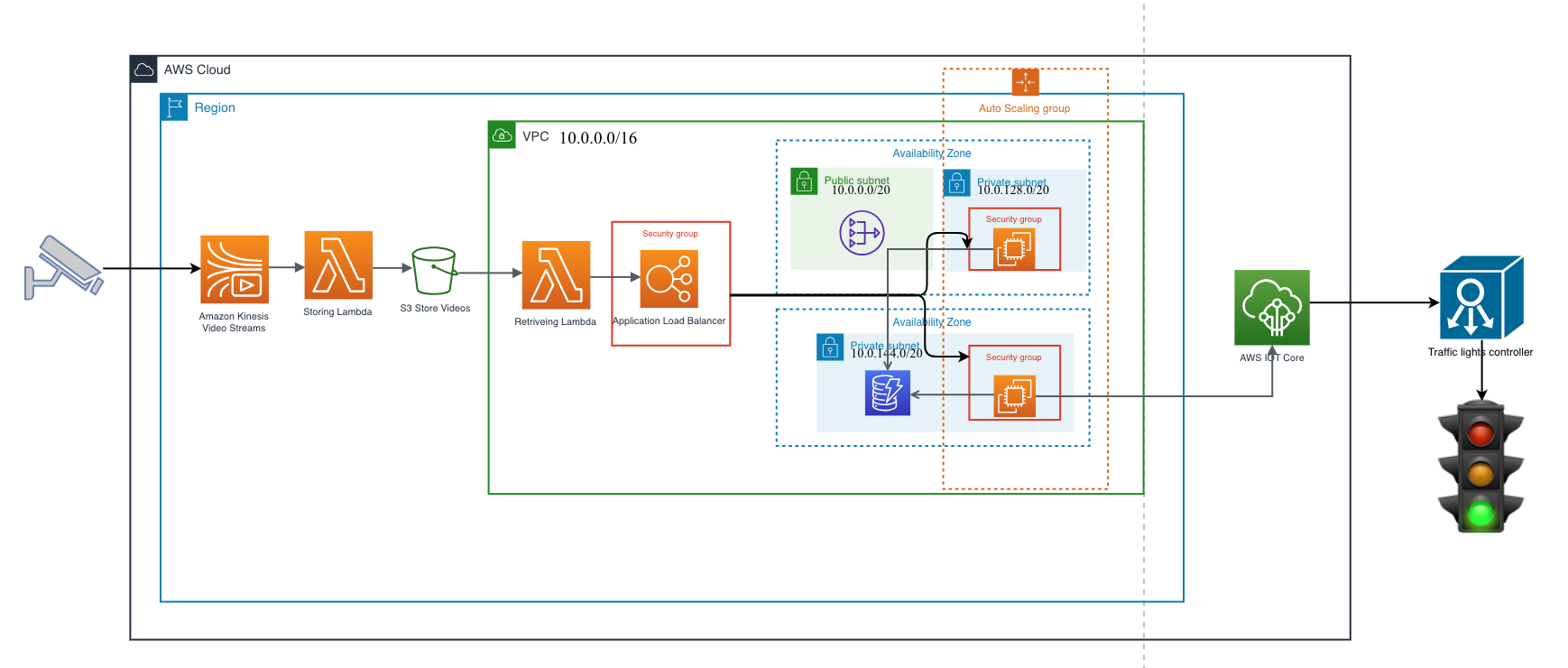
­­­­­

Figure 1. 1: Smart Traffic Management System architecture

# The figure above shows the infrastructure for the Smart Traffic Management (STM) system. This architecture focuses on implementing a highly available, reliable, scalable, and secure traffic management system leveraging AWS services. The architecture consists of a VPC that contains two availability zones.

# Amazon Kinesis Video Streams

# This service is used to collect real-time video data from traffic cameras. It is chosen for its ability to handle large volumes of streaming data and integrate seamlessly with other AWS services for processing.

# Storing Lambda

# This AWS Lambda function storing the videos in Amazon S3. It ensures that the system remains serverless, reducing the operational overhead.

# Amazon S3

# S3 is used to store the video data collected from cameras. Its scalability and durability make it an ideal choice for storing large amounts of data. Additionally, it supports secure data transfer via HTTPS, ensuring data security during transmission.

# Retrieving Lambda

# Another AWS Lambda function is used to retrieve videos from S3.

# Network Load Balancer

# The load balancer distributes incoming traffic across multiple instances, ensuring high availability and fault tolerance. It is essential for managing the traffic load efficiently and preventing any single point of failure.

# Auto Scaling Group

# This group automatically adjusts the number of EC2 instances based on traffic load. It ensures that the system can handle varying loads efficiently, scaling up during peak hours and scaling down during off-peak times to optimize costs.

# Amazon EC2 Instances

# EC2 instances are used to host the applications that process and analyze the traffic videos and data. These instances are placed in private subnets to enhance security and prevent direct access from the internet.

# Private and Public Subnets

# The architecture includes both private and public subnets within the first availability zone and one public subnet within the second availability zone. Private subnets host the EC2 instances, while public subnet contain the NAT gateway, ensuring secure and efficient communication within the VPC.

# NAT Gateway

# This gateway allows instances in the private subnets to connect to the internet securely, enabling software updates and data transfers without exposing the instances directly to the internet.

# Amazon DynamoDB

# DynamoDB is used to store historical traffic data, incident reports, and system logs. Its automatic replication across multiple availability zones and since it’s NoSQL it can handle massive read and write rates, which ensures high availability and fault tolerance, making it suitable for storing critical data.

# AWS IoT Core

# This service manages the communication between traffic cameras and the cloud, allowing for real-time data collection and processing. It is chosen for its ability to handle a large number of connected devices securely and efficiently.

# This comprehensive infrastructure design ensures that the STM system is robust, scalable, and secure, leveraging AWS services to optimize traffic management in urban areas.

# Procedure:

## **VPC**

The first step for building the infrastructure is creating a VPC with address (10.0.0.0/16) containing two availability zones. Availability zone 1 contains one public subnet (10.0.0.0/20) and one private subnet (10.0.128.0/20). Availability zone 2 also contains one public subnet (10.0.16.0/20) and one private subnet (10.0.144.0/20). The instances are distributed across the two availability zones for higher availability and fault tolerance.

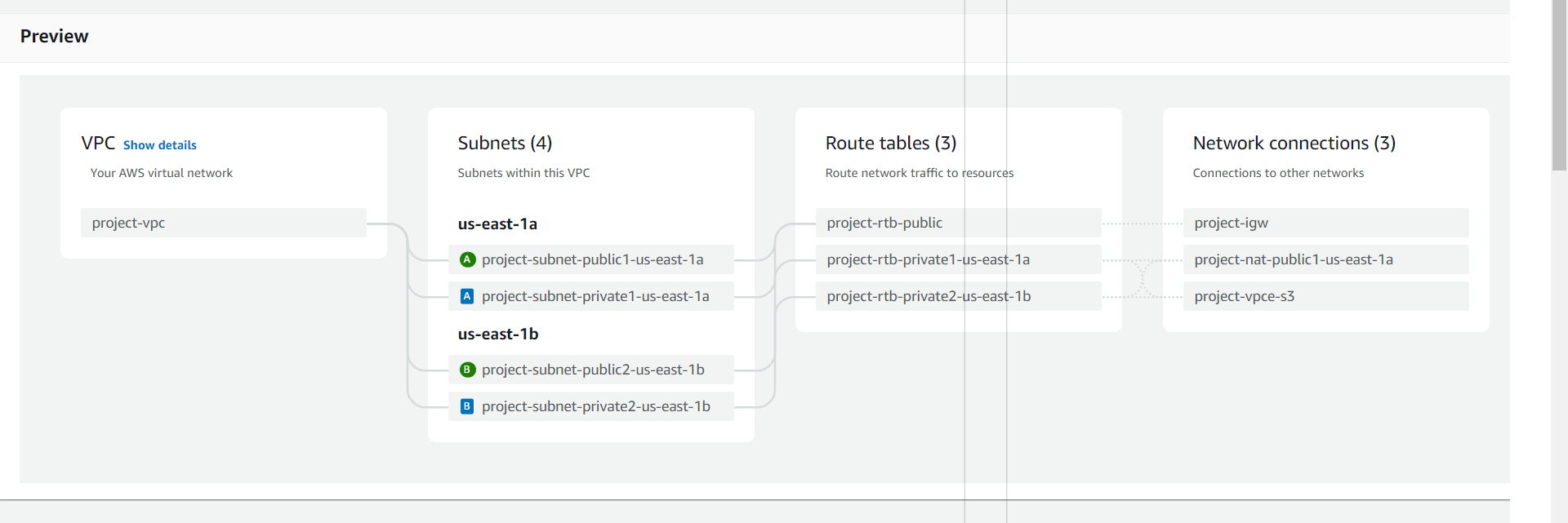


Figure 1.2: VPC resource map

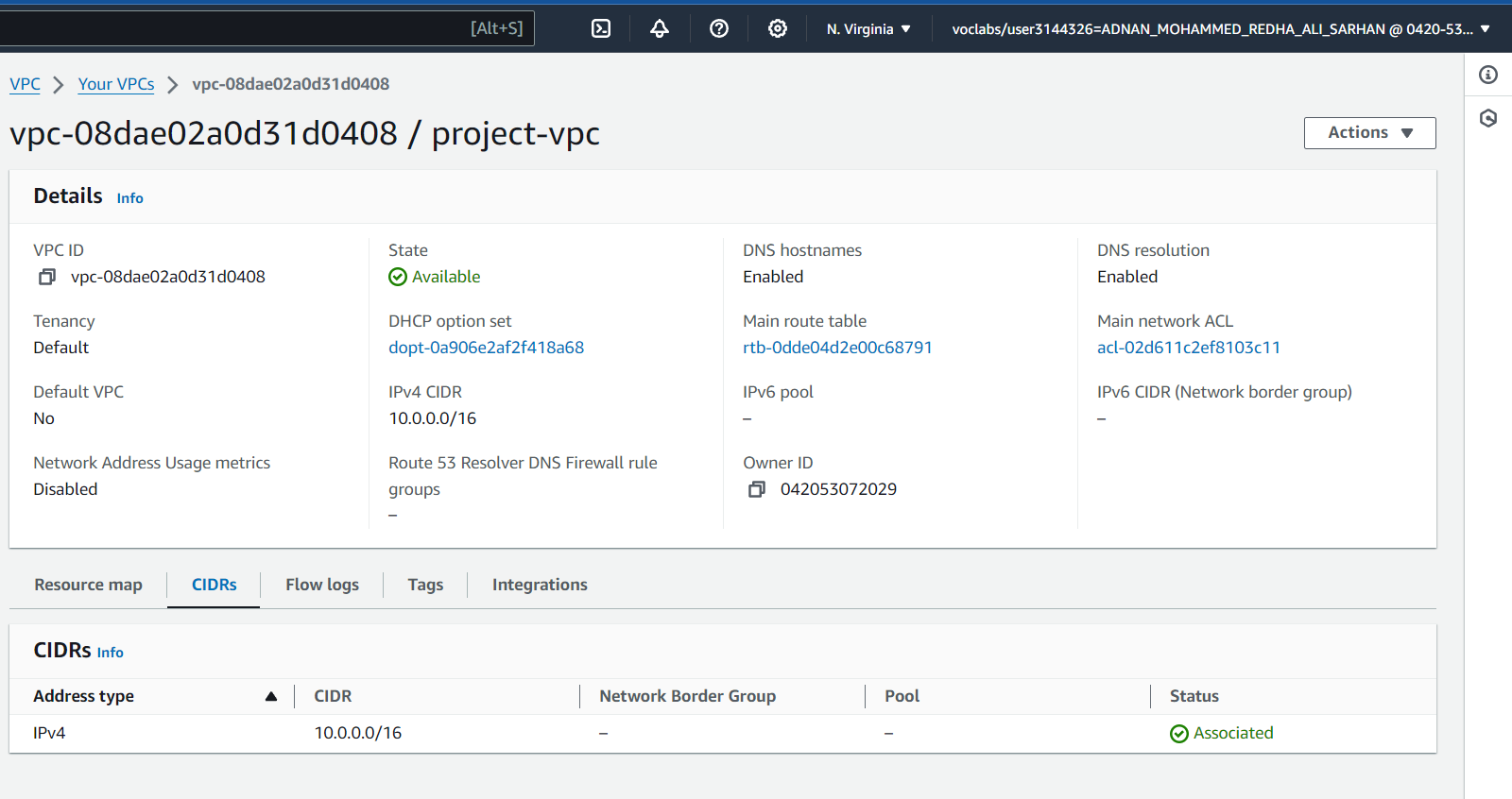


Figure 1. 3: VPC Details

Figure 1. 4: Subnets Details

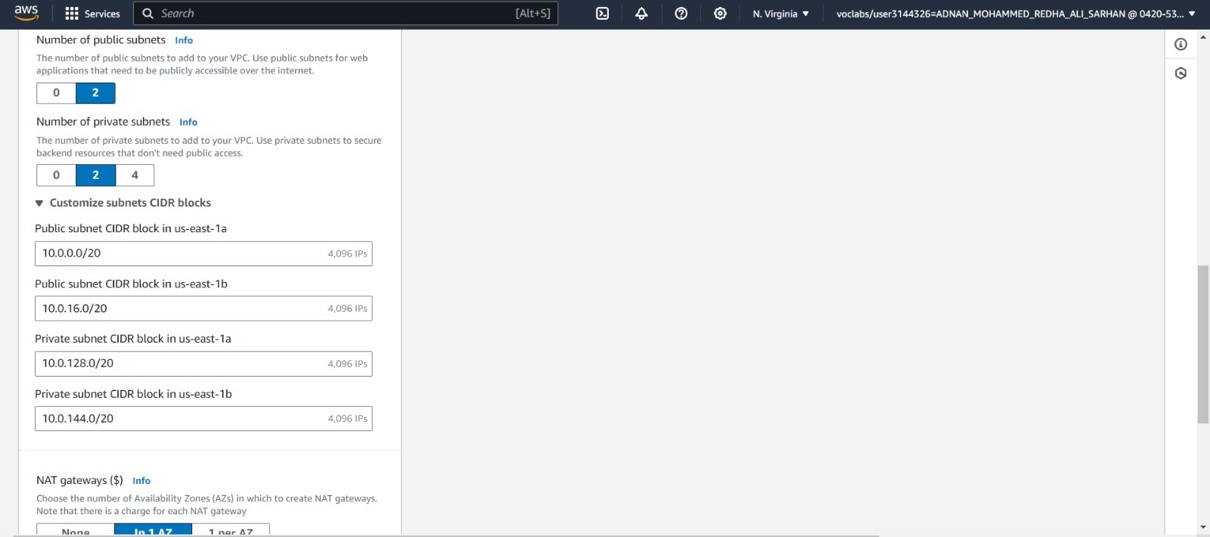


Figure 1. 4: Subnets Details

## **NAT**

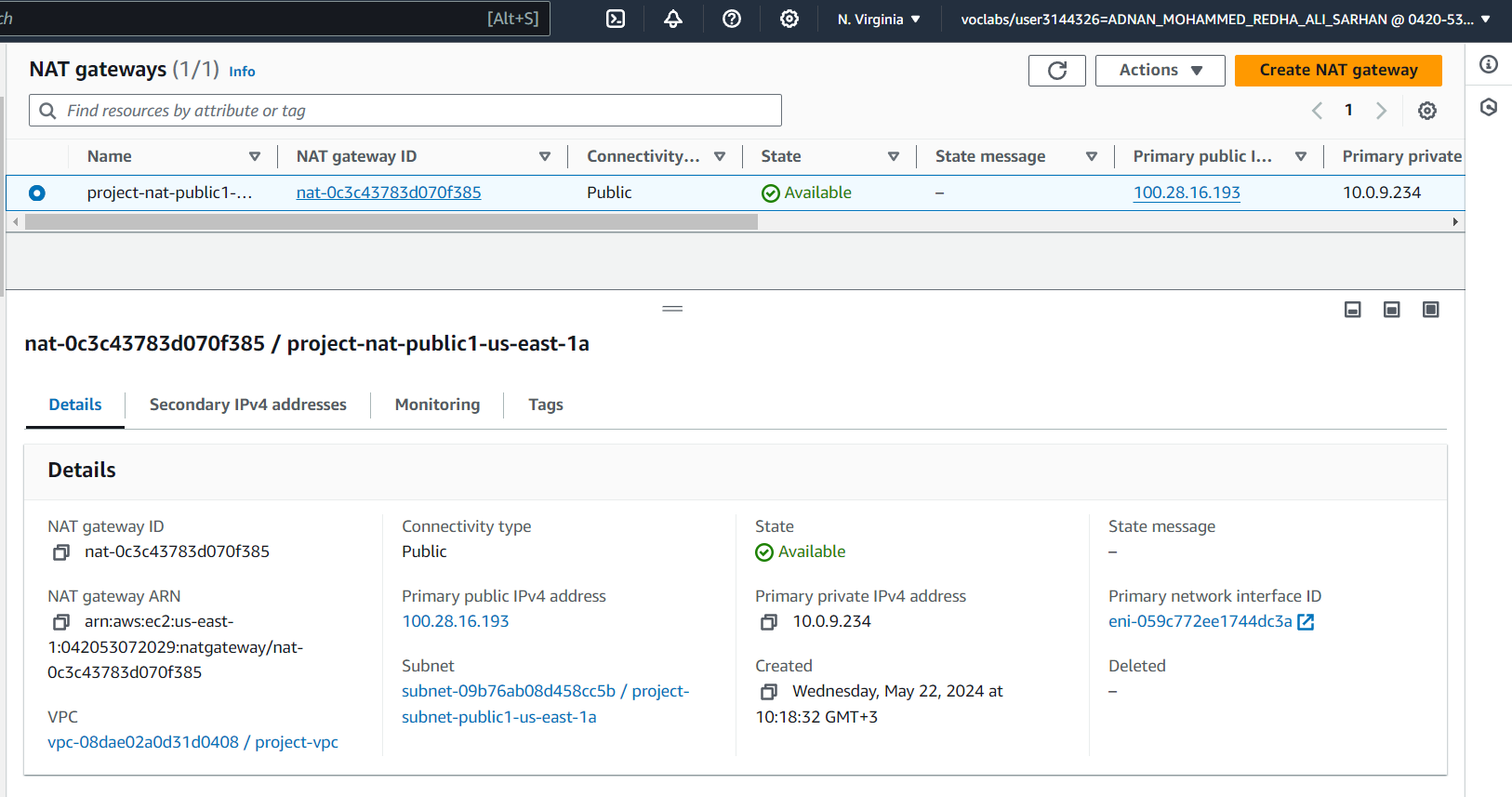
A Network Address Translation (NAT) gateway link between private and public networks, allowing the private network a secure connection to the internet. NAT gateways enable instances in the private subnets to get access to the internet securely.

Figure 1. 5: NAT gateway details

## **Routing Table**

The routing table contains routes (set of rules) which specifies target and destination for each route, and it controls the inbound and outbound traffic for a subnet. Two routing tables were created for both public subnets with the same configurations. They contain a route to the NAT beside the local route. Also, two routing tables for the two private subnets were created

with the same configurations, having local route, route to the NAT gateway, and a route for the load balancer.

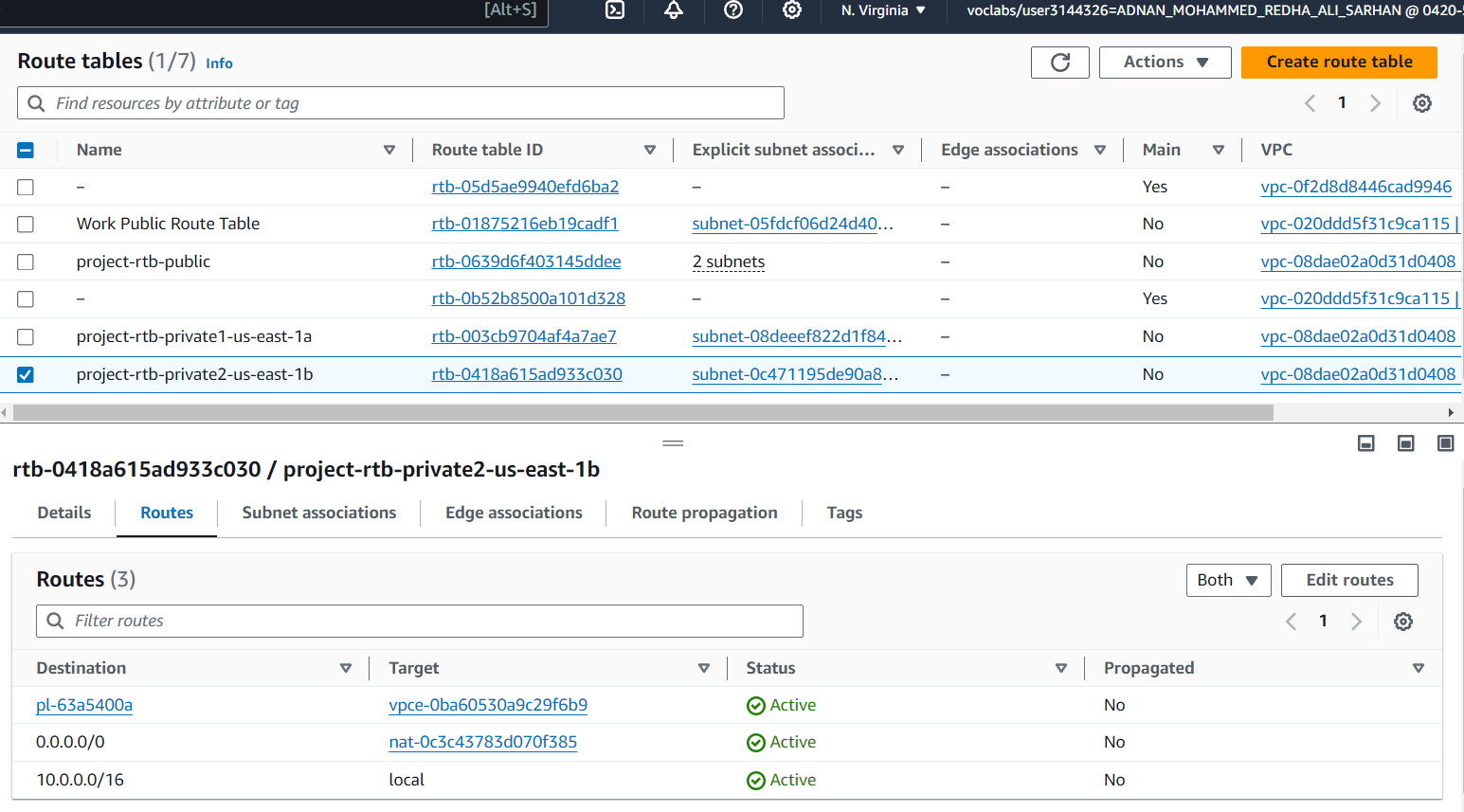


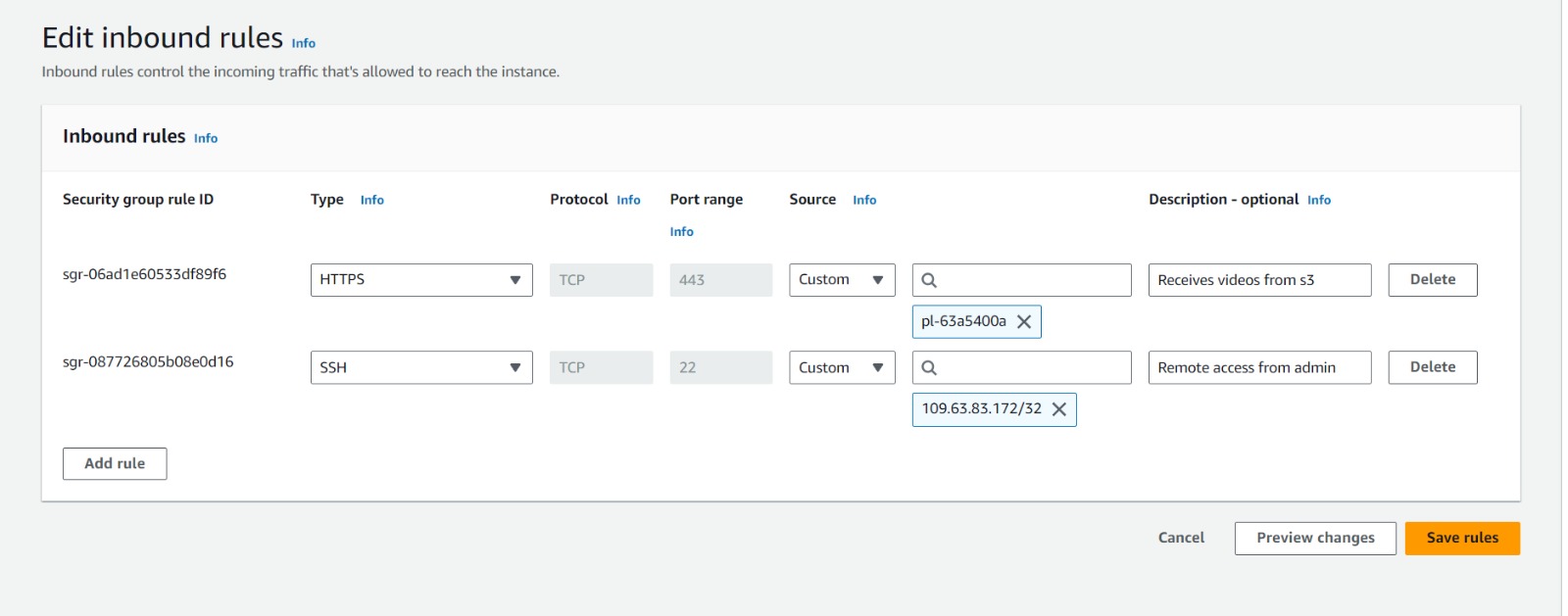
Figure 1. 6: Routing table details

with the same configurations, having local route, route to the NAT gateway, and a route for the load balancer.

## **Security Groups**

Security groups act as a virtual firewall that manages inbound and outbound traffic. We added security groups for the load balancer and the two EC2 instances. We allowed all outbound traffic and restricted the inbound traffic as shown in the figures.

### **Load Balancer**

Figure 1. 7: Load Balancer’s Security Group details

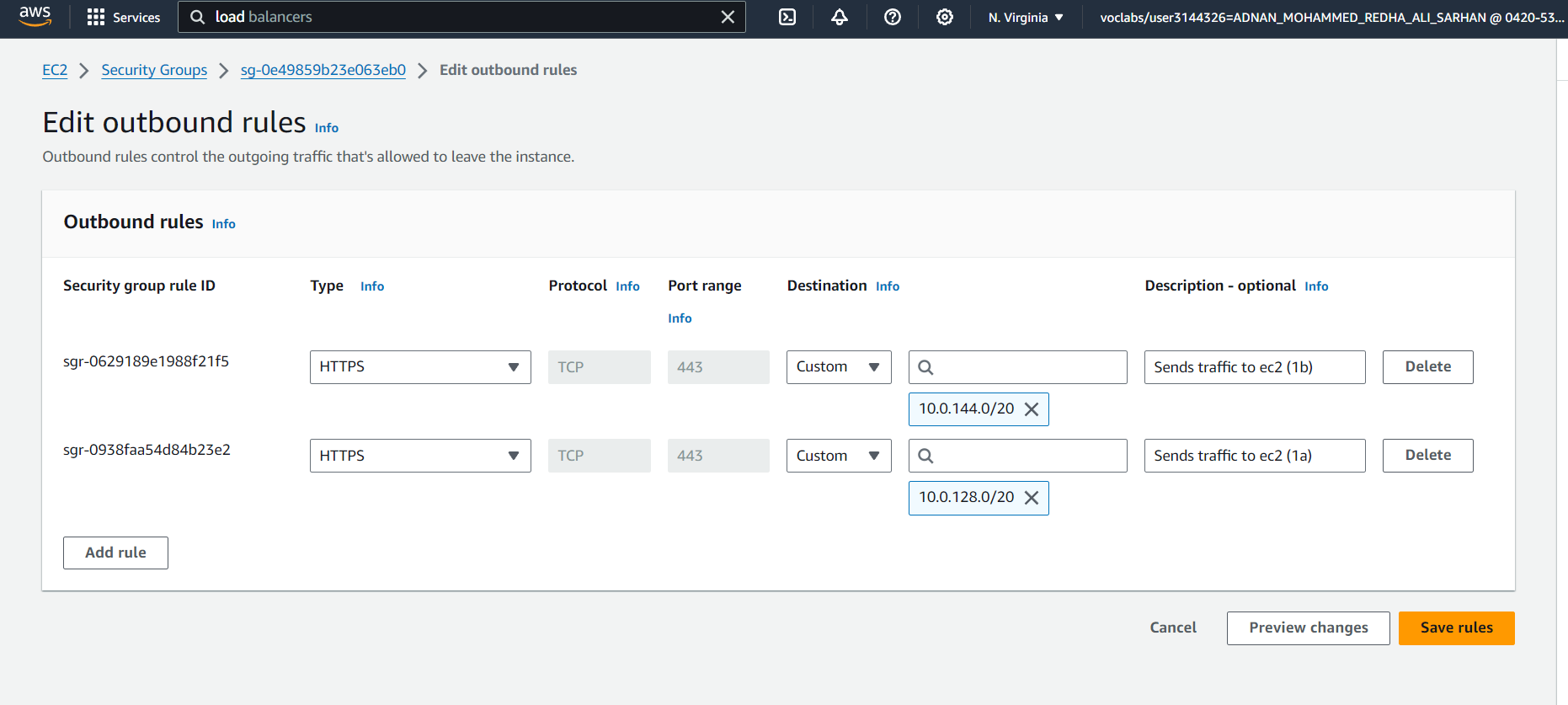
Figure 1. 8: Load Balancer’s Security group inbound rules

Figure 1. 9: Load Balancer’s Security group outbound rules

### **EC2**



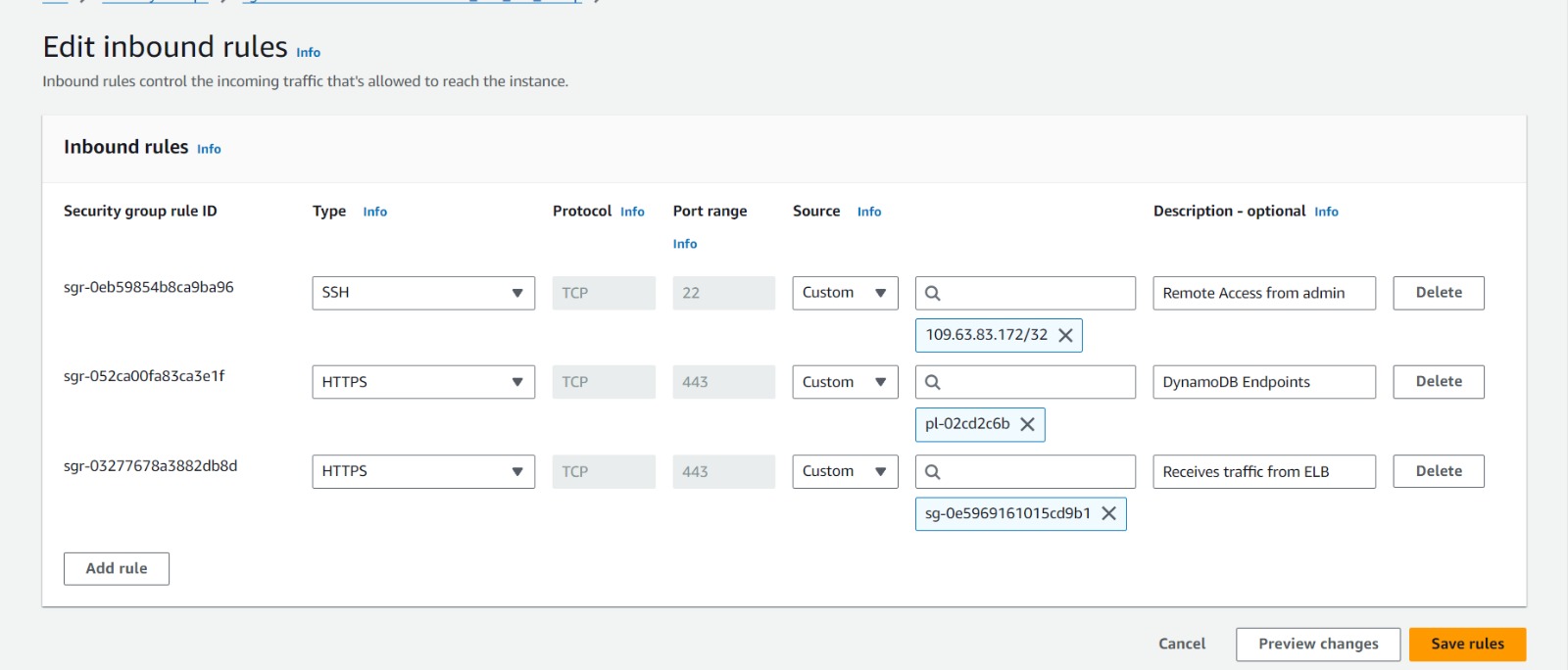
Figure 1. 10: EC2 instances’ Security Groups Details

Figure 1. 11: EC2 instances’ security groups inbound rules

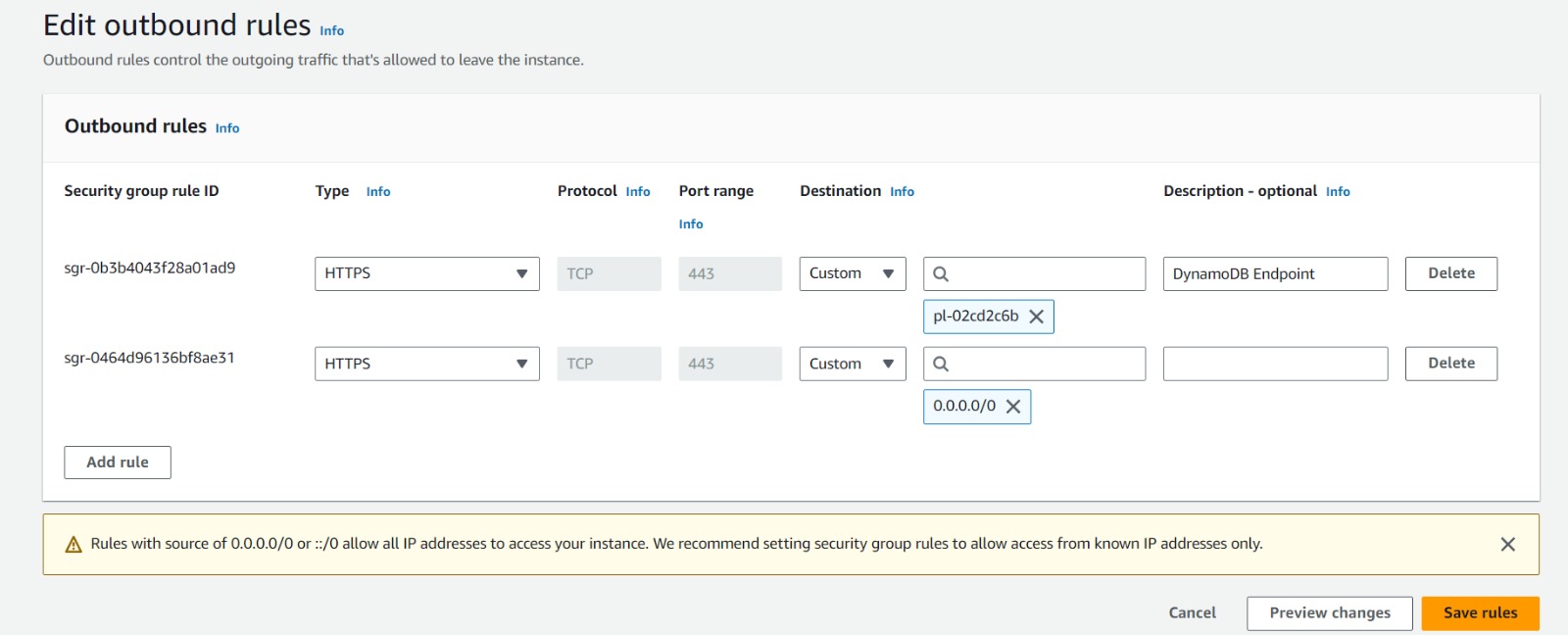


Figure 1. 12: EC2 instances’ security groups outbound rules

1. **Instances**

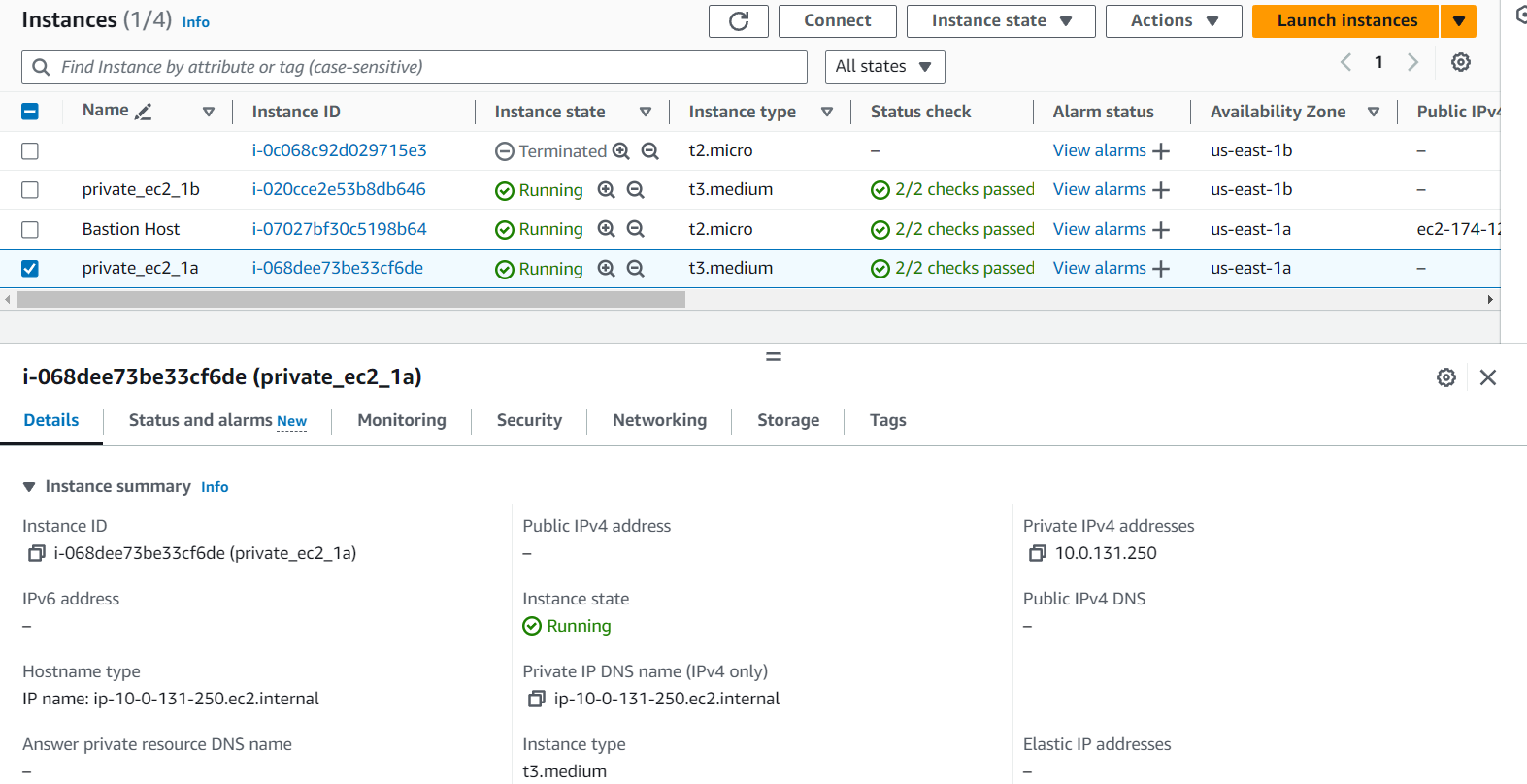
two web servers were created to process the videos.

Figure 1. 13: Project Instances

## **DynamoDB**

The traffic decisions made must be stored in DynamoDB, the results of the machine learning model and a reference to the video stored in S3 are retained here. The aim of performing such a task is to have previous decisions stored for any retraining necessary that might occur to the model. However, when trying to create a DynamoDB table, the following error keeps popping up.

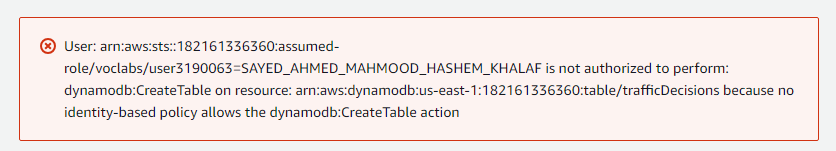
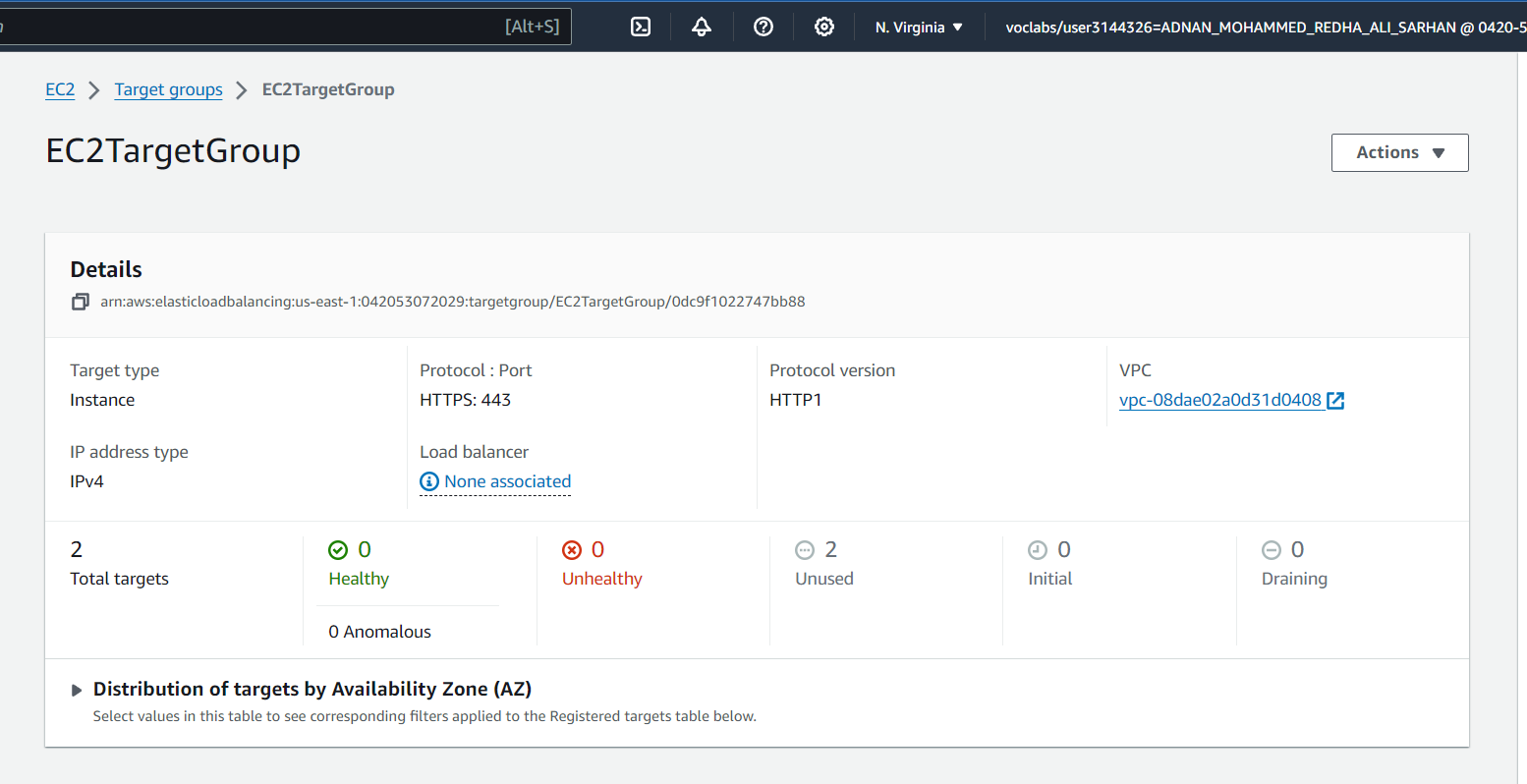
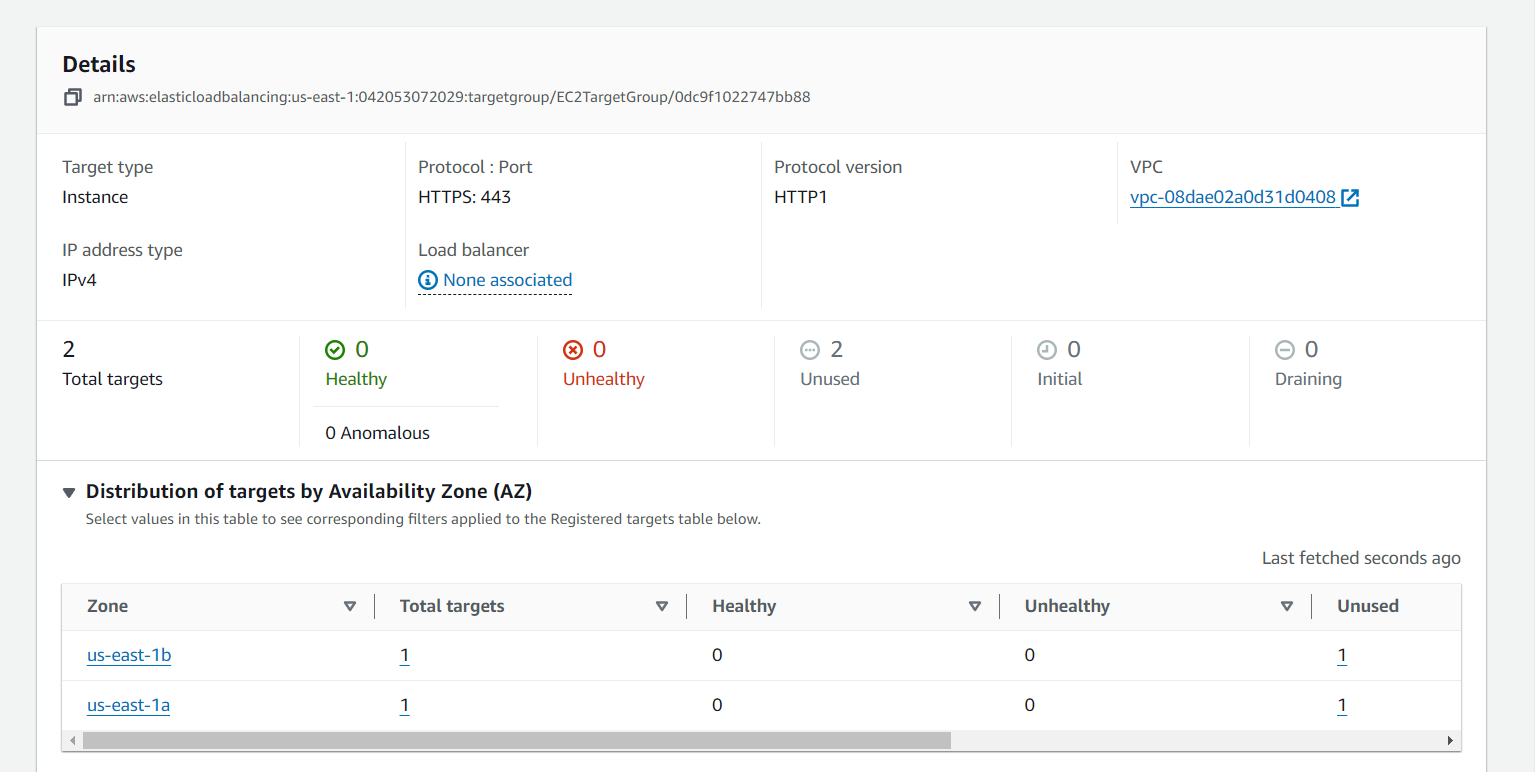


Figure 1. 14: DynamoDB

## **Target groups and load balancers**

with the same configurations, having local route, route to the NAT gateway, and a route for the load balancer.





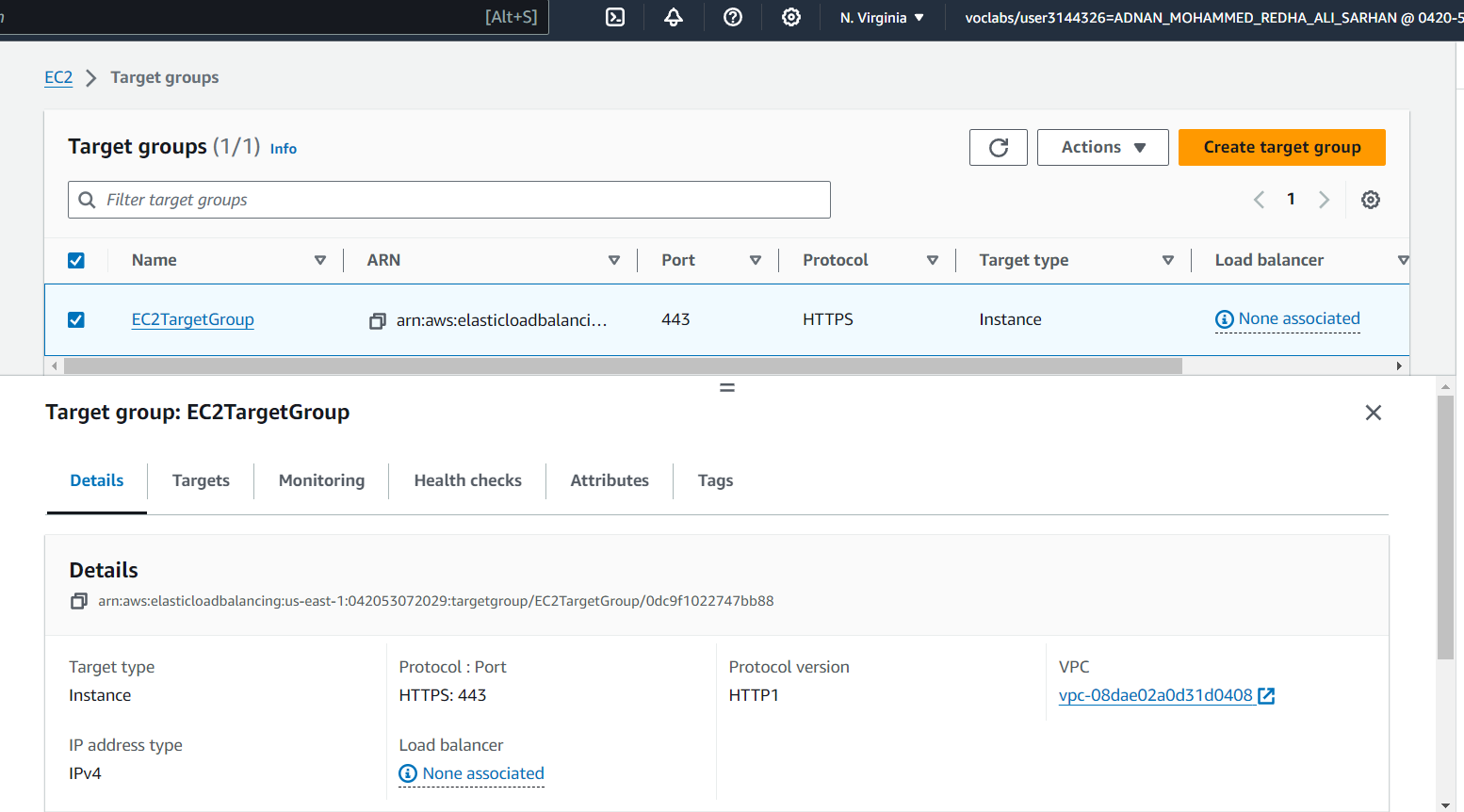


Figure 1. 15: Target group

1. **S3 Bucket**

The S3 bucket contains traffic videos that are sent from the cameras every 15 seconds since we need continuously new data.

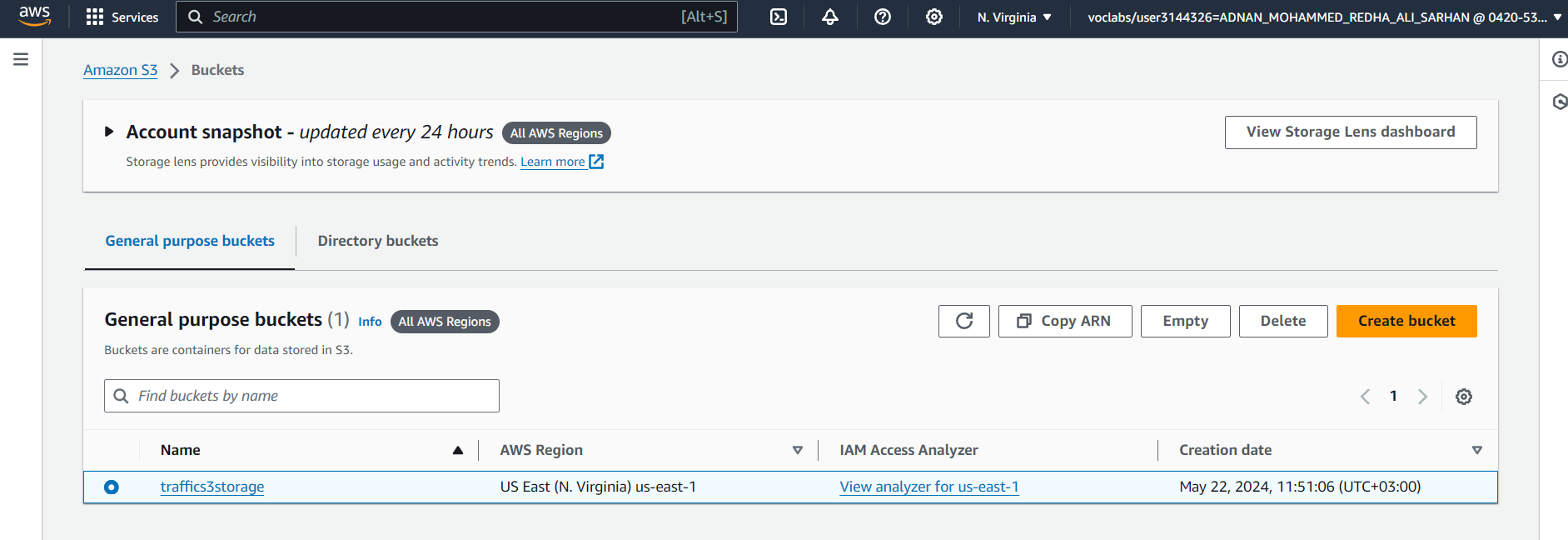
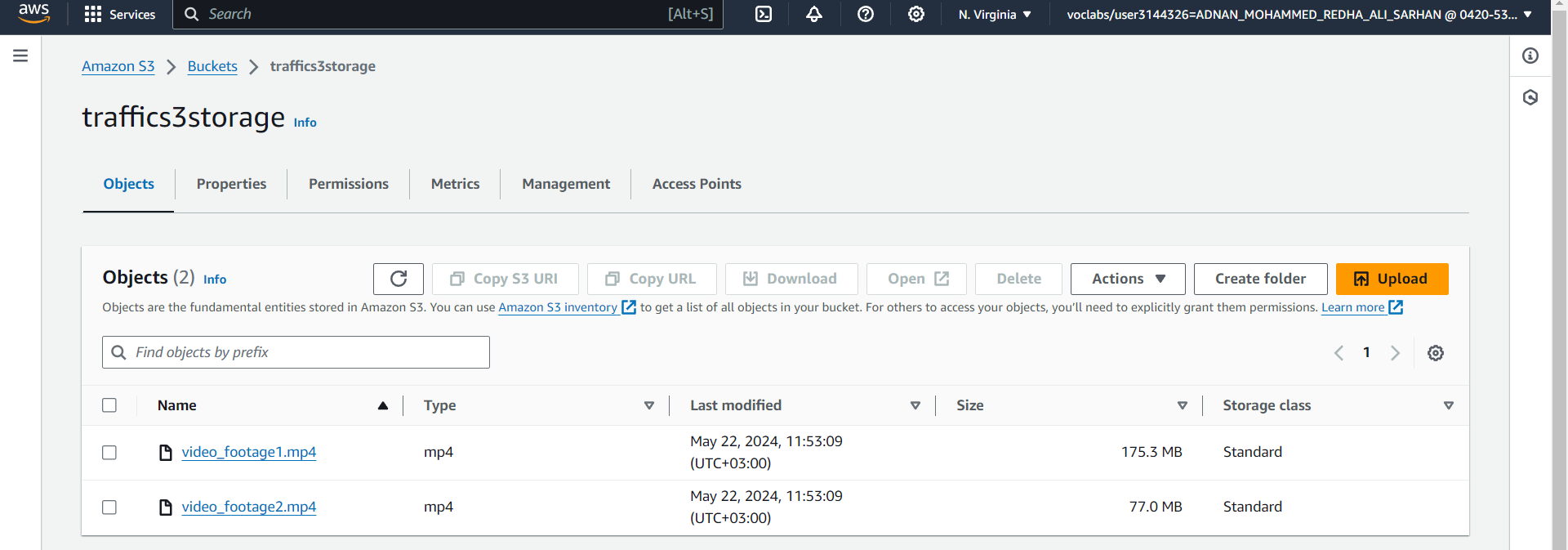
****

Figure 1. 16: S3 Bucket

## **AMI**

An image was created from the EC2 instances for the purposes of reliability and efficiency in the case of any failures.

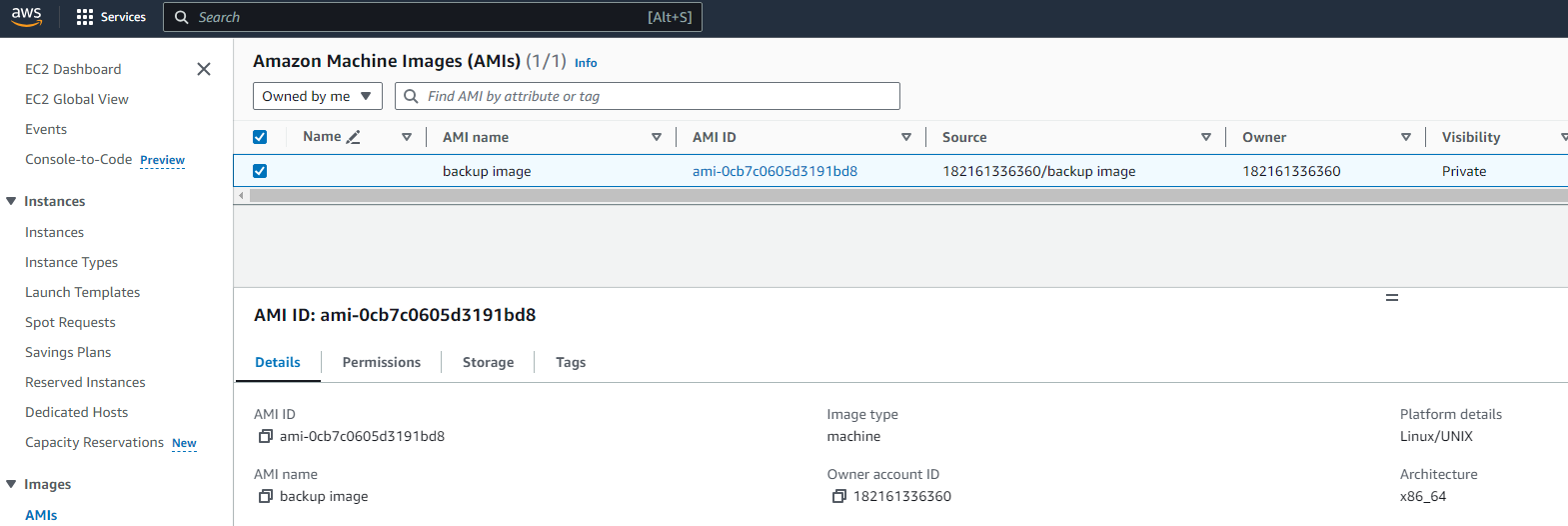


Figure 1. 17: AMI

## **Auto-scaling**

## An Auto Scaling group can be thought of as a collection of EC2 instances grouped together logically for the goal of automatic scaling. Auto Scaling group lets you use health check replacements and scaling policies based on your needs. The size of the Auto Scaling group can be adjusted to meet demand, and this can be reflected through the indication of the desired capacity.

## 

Figure 1. 18: Auto-Scaling

# Cost Estimation

The estimated cost for implementing the Smart Traffic Management (STM) system based on the AWS Pricing Calculator is as follows. The upfront cost is $180.00 USD, with a monthly cost totaling $910.02 USD. Considering a 12-month timeframe, the total cost, including the upfront cost, amounts to $11,100.24 USD.

The cost breakdown for different AWS services is as follows:

Amazon Simple Storage Service (S3): The monthly cost for S3 in the US East (N. Virginia) region is estimated at $235.52 USD. This includes 10 TB of S3 Standard storage used per month.

Amazon Kinesis Video Streams: The monthly cost for Kinesis Video Streams in the US East (N. Virginia) region is estimated at $222.53 USD. This estimate takes into account the configuration of 20 devices, each streaming video at an average bitrate of 1.2 Mbps for 24 hours per day.

AWS Lambda: No monthly cost is estimated for AWS Lambda, assuming it to be $0.00 USD. The configuration assumes an x86 architecture, buffered invoke mode, and 512 MB ephemeral storage allocation. The estimate is based on 900,000 requests per month.

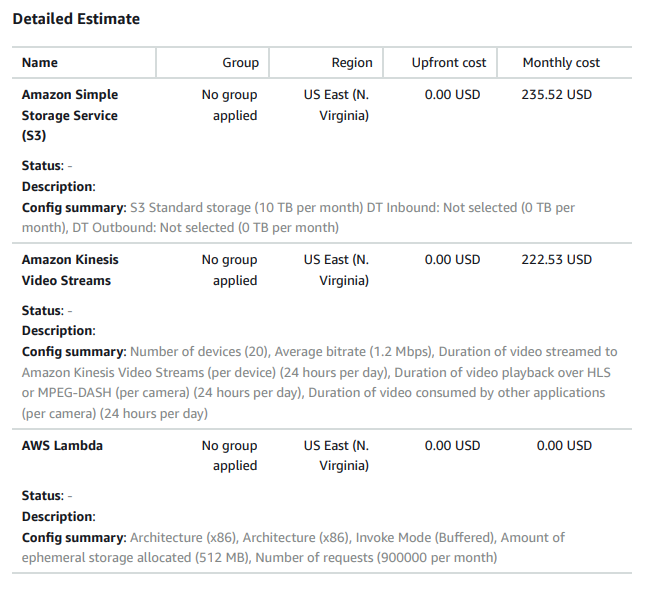
Amazon EC2: The monthly cost for EC2 in the US East (N. Virginia) region is estimated at $34.28 USD. The configuration includes two instances of the t3.medium type with shared tenancy, running Linux, and utilizing Compute Savings Plans for a 3-year term.

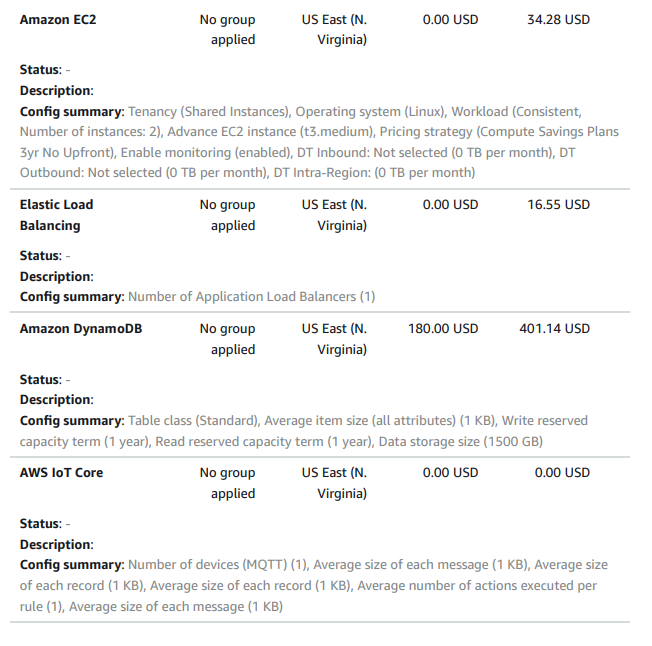
Elastic Load Balancing: The monthly cost for Elastic Load Balancing in the US East (N. Virginia) region is estimated at $16.55 USD, assuming the presence of one Application Load Balancer.

Amazon DynamoDB: The upfront cost for DynamoDB in the US East (N. Virginia) region is $180.00 USD, with a monthly cost of $401.14 USD. The configuration includes a standard table class, an average item size of 1 KB, and reserved capacity terms for both write and read operations.

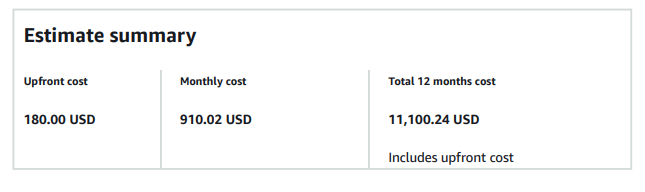
AWS IoT Core: No monthly cost is estimated for AWS IoT Core, assuming it to be $0.00 USD. The configuration includes one MQTT device, with an average size of 1 KB for each message and record.

In summary, the cost estimation provides a breakdown of the anticipated costs for the various AWS services involved in implementing the Smart Traffic Management (STM) system.





*Figure 1. 29: AWS used services cost estimation*

*Figure 1. 30: Cost estimate summary*

# Conclusion

The Smart Traffic Management (STM) system leverages AWS services to create an efficient, scalable, and secure solution for urban traffic management. By dynamically adjusting traffic signal timings based on real-time data, the system aims to reduce congestion, improve mobility, and minimize environmental impact. The designed infrastructure ensures high availability, fault tolerance, and robust data security. Cost estimation indicates that the system is economically viable for large-scale implementation, providing a comprehensive approach to enhancing urban traffic flow.

When it comes to future work, using AWS-managed services would reduce the overhead related to managing EC2 instances. Amazon Rekognition is a good contender for the analysis of the video streams, as an alternative to our choice of EC2 instances. However, we cannot mention Rekognition without describing some of the limitations associated with it. Using Rekognition would give us less control over the choice of ML model for analyzing traffic congestion. Furthermore, it gives us less flexibility for any future enhancements. A more novel approach would be to utilize edge computing for analyzing the video streams. This can be achieved by utilizing another AWS service called Panorama. Following this approach reduces the traffic by a sizeable margin and ultimately enables more efficient and cost-effective scalability.