

## AWS CLOUD SOLUTIONS PROJECT 2

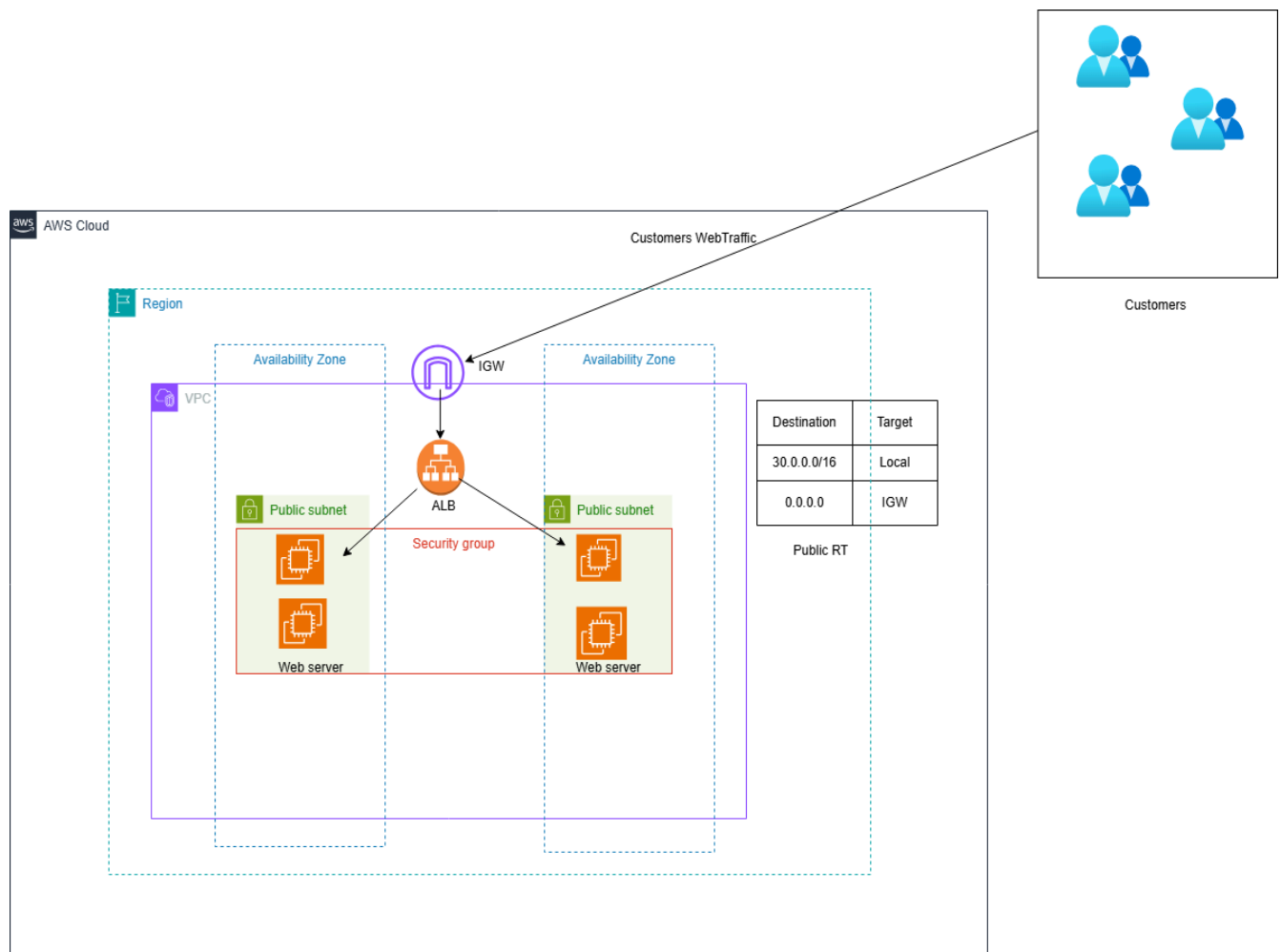
### Setting Up An Application Load Balancer On AWS Cloud.

This project shows a detailed step-by-by process of setting up an application load balancer on AWS cloud.

This project validates my knowledge and skills in the following areas on AWS cloud.

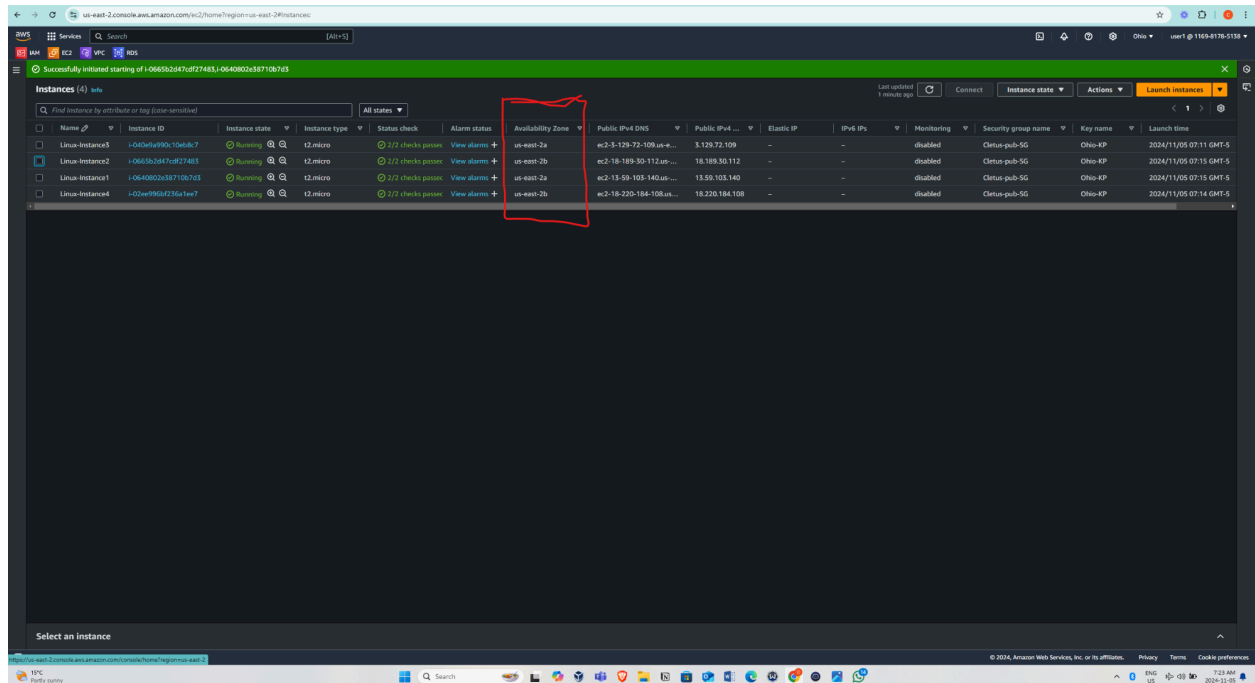
1. AWS infrastructure (Load balancer)
2. IP Address (Cidr block)
3. Subnetting
4. Virtual Machines (EC2)
5. Linux
5. Network Traffic
6. Security (SG and NACL)

Below is the architectural diagram of this project:



## Step 1

Spinning up four linux servers in 2 availability zones. Two in each of the availability zones. This is for the purpose of ensuring high availability of the application that would be hosted on these servers.



## Step 2

Connect to each instance via SSH and install the Apache2 web server. Once the installation is complete, I will modify the index.html file on each server to include unique content for easy identification. This will help differentiate the servers when accessing them through an application load balancer.

```

[ec2-user@ip-20-0-0-105 ~]$ sudo systemctl status httpd
systemd operation: success
[ec2-user@ip-20-0-0-105 ~]$ sudo systemctl status httpd
systemd operation: success
Loaded loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
Active: active (running) since Tue 2024-11-05 14:53:05 UTC; 23s ago
Docs: man:httpd.service(8)
Main PID: 9808 (httpd)
Status: "Total requests: 0; idle/busy workers: 100/0;requests/sec: 0; Bytes served/sec: 0 B/sec"
CGroup: /system.slice/httpd.service
┌─9808 /usr/sbin/httpd -DFOREGROUND
┌─9809 /usr/sbin/httpd -DFOREGROUND
┌─9878 /usr/sbin/httpd -DFOREGROUND
┌─9872 /usr/sbin/httpd -DFOREGROUND
┌─9872 /usr/sbin/httpd -DFOREGROUND
└─9873 /usr/sbin/httpd -DFOREGROUND

Nov 05 14:53:05 ip-20-0-0-105.us-east-2.compute.internal systemd[1]: Starting The Apache HTTP Server....
Nov 05 14:53:05 ip-20-0-0-105.us-east-2.compute.internal systemd[1]: Started The Apache HTTP Server.
[ec2-user@ip-20-0-0-105 ~]$

-- INSERT --
```

```

<!-- DOCTYPE html -->
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Server 1</title>
  <style>
    body {
      margin: 0;
      font-family: 'Arial', sans-serif;
      background-color: #f5f5f5;
      color: #333;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
    }
    header {
      font-size: 36px;
      font-weight: bold;
      text-align: center;
      margin-bottom: 20px;
    }
    main {
      max-width: 800px;
      padding: 20px;
      background-color: #ffffff;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      border-radius: 8px;
    }
  </style>
</head>
<body>
  <header>This is server 2</header>
  <main>
    <p>Welcome to the static website hosted on server 2! This site serves as a demonstration of a simple yet elegant static webpage design. Feel free to modify the content to fit your needs.</p>
  </main>
</body>
</html>
<!--
```

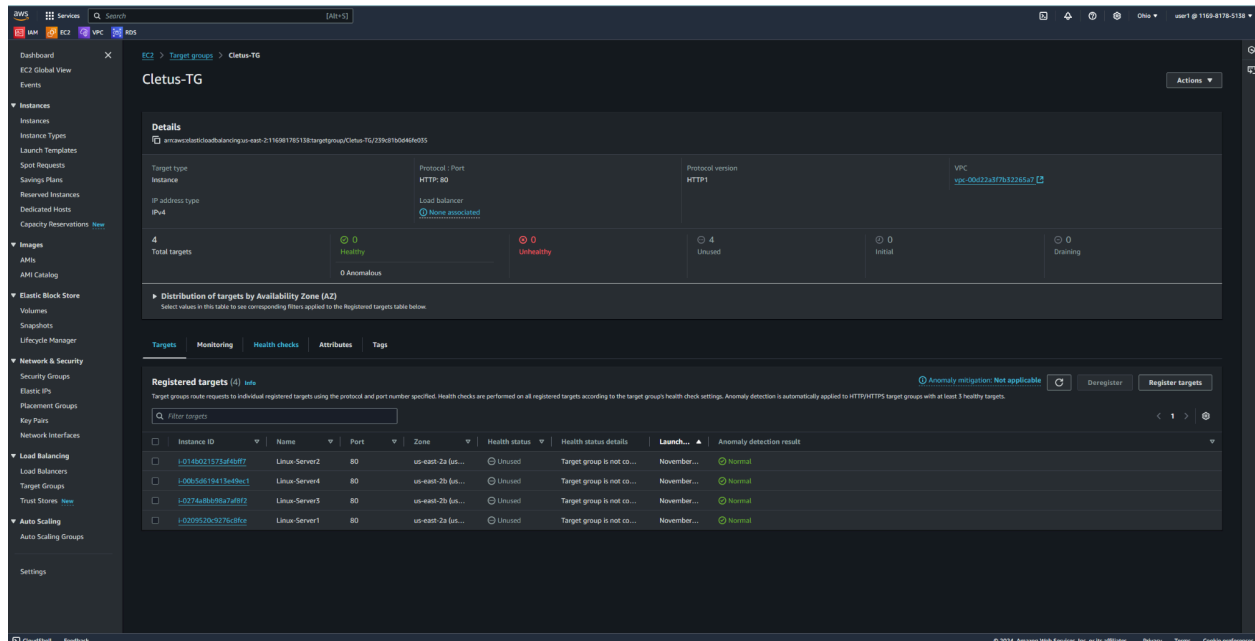
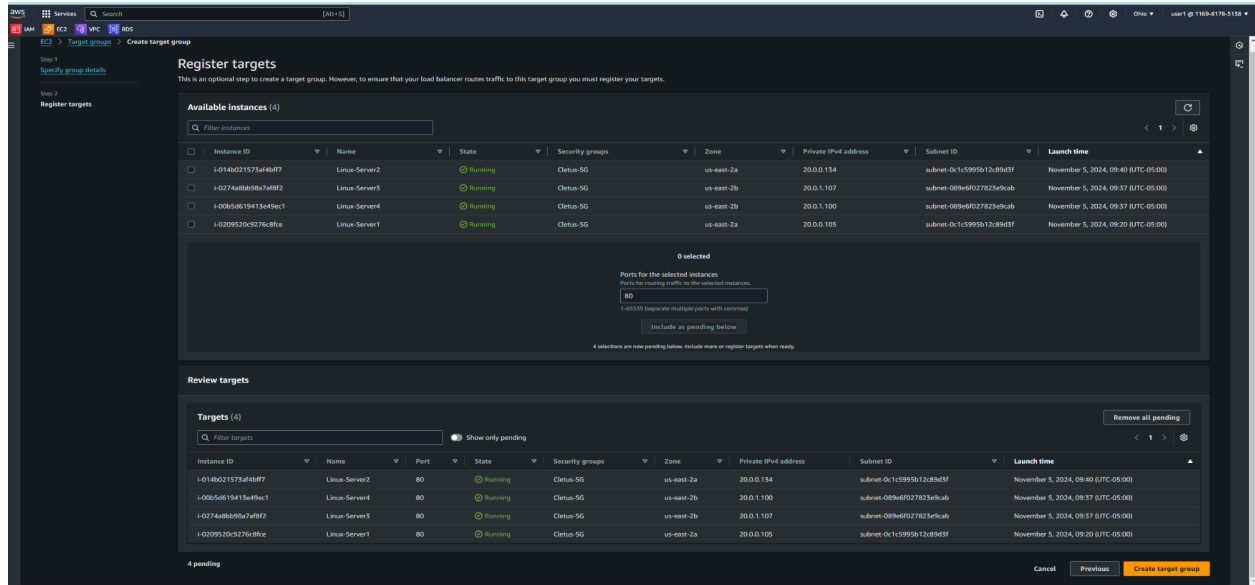
```

## Explanation:
- <!--HTML Structure-->: The page is structured with a 'header' element to display the bold title and a 'main' section for additional content.
- <!--CSS Styling-->:
  - 'body' is styled to center the content on the page, with a clean font and background.
  - 'header' has a larger font size and bold styling to emphasize the title.
  - 'main' is styled with a white background, rounded corners, and a subtle shadow for a modern card-like effect.

You can save this code in an 'index.html' file and place it in your web server's root directory (e.g., '/var/www/html/') for Apache
-- INSERT --
```

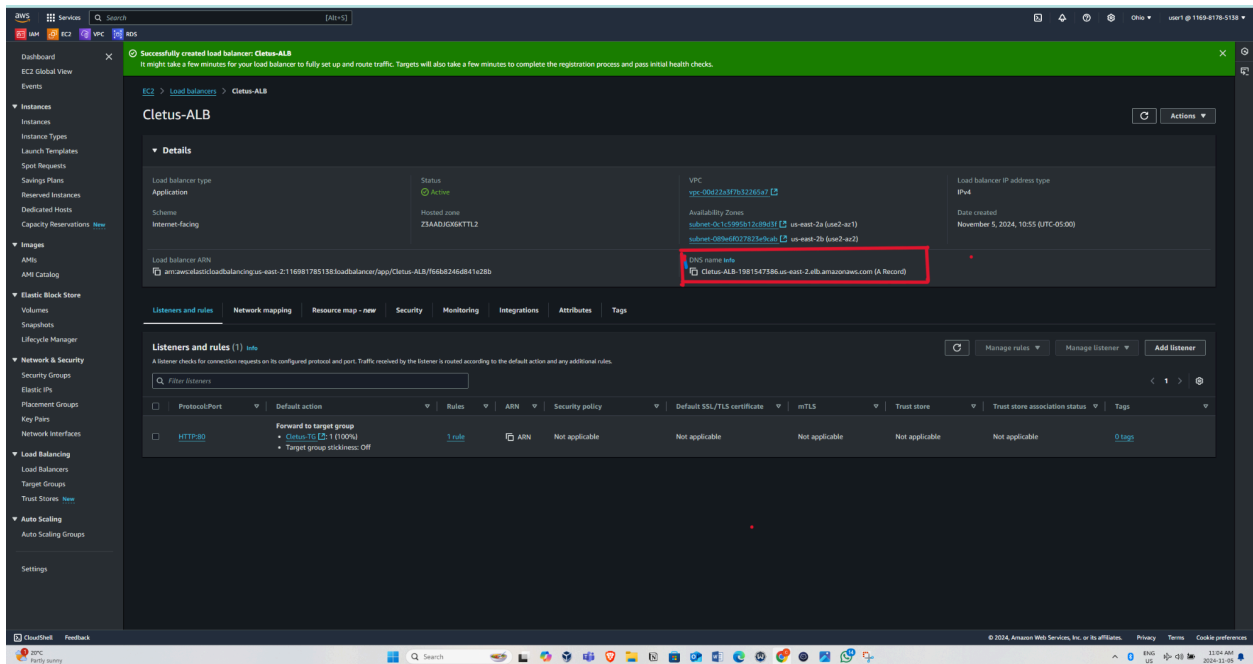
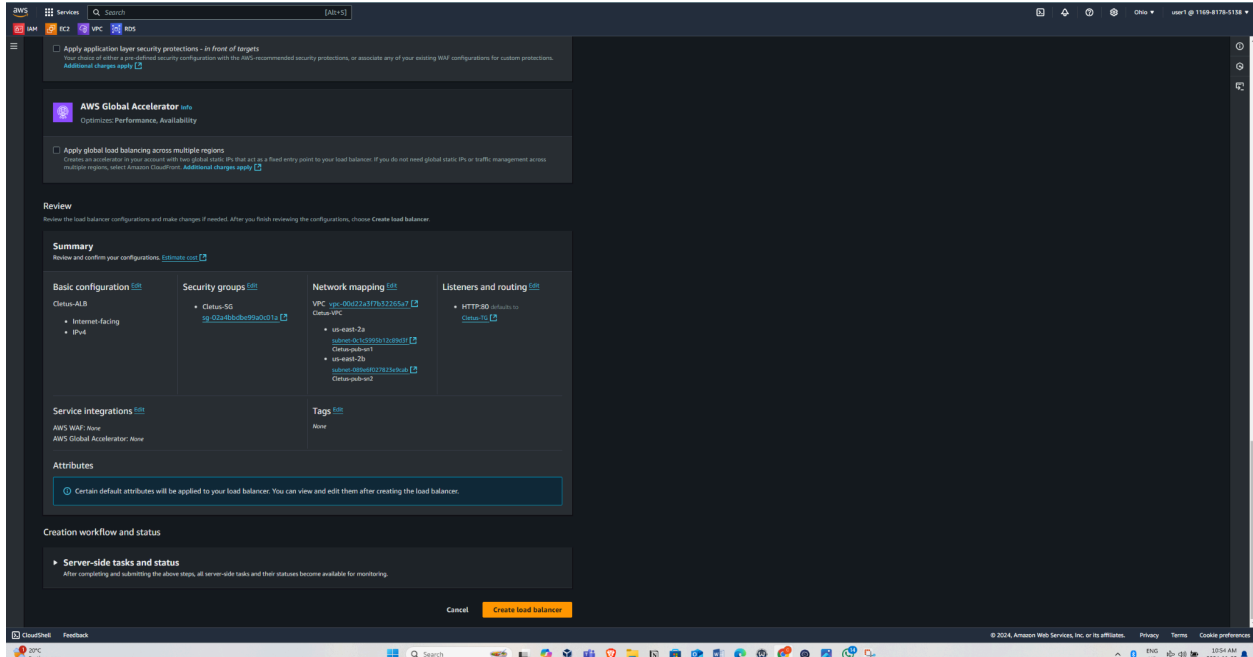
# Step 3

Create a target group. Target group is a group of instances or servers that the load balancer would direct incoming traffic to based on specified configurations. So I will be putting the four servers I have created in one target group.



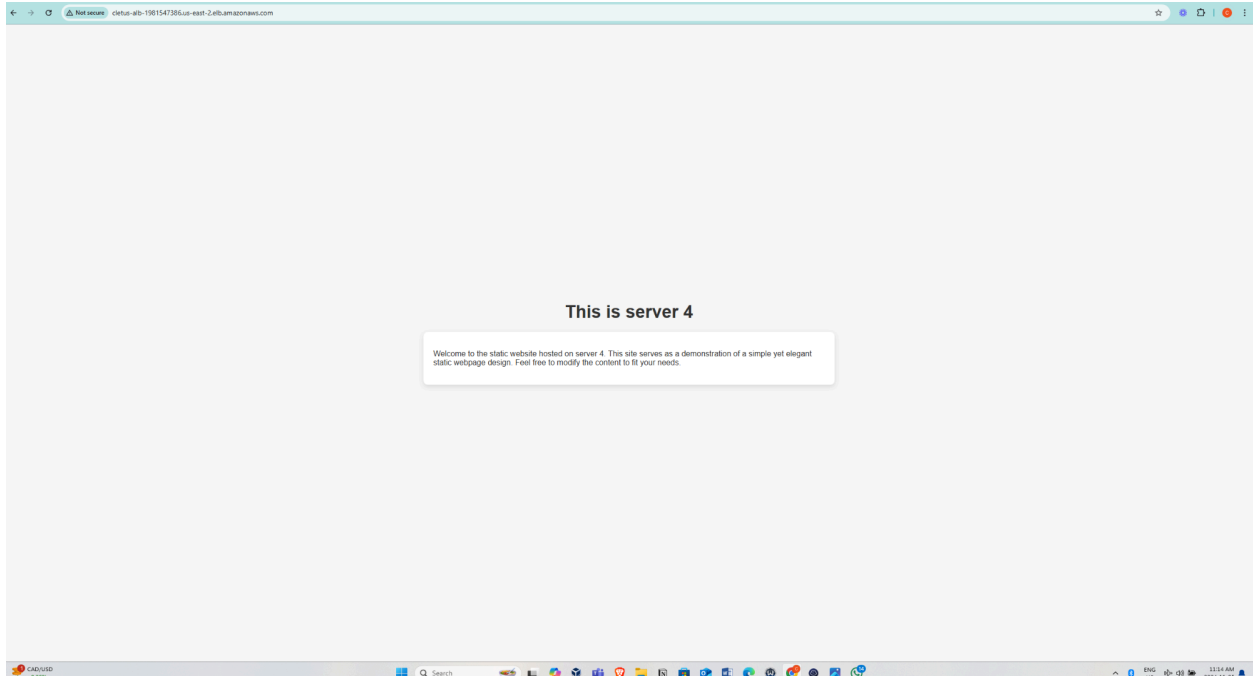
## Step 4

Create an Application Load balancer and associate it with the target group I have created. This will ensure that incoming traffic directed to the load balancer would be routed to the servers within the target group.



## Step 5

Browsing for the DNS A record of the load balancer to verify what I have done so far.



Every time I refresh, I receive a response from a different server. This confirms that the four servers placed behind the load balancer are functioning properly and effectively handling incoming traffic.

