Implementing a Load-Balanced Calculator Application in AWS

Contents

1 Overview

This document outlines the steps to deploy a simple calculator web application across two AWS availability zones using an Application Load Balancer (ALB) for high availability and scalability. The application is built with HTML, CSS, and JavaScript, hosted on EC2 instances.

2 Prerequisites

- AWS account with permissions to create VPC, EC2, ALB, and related resources.
- Basic knowledge of AWS services, HTML, CSS, and JavaScript.
- AWS CLI configured on your local machine.
- SSH key pair for EC2 access.

3 Calculator Application Code

3.1 HTML (index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Calculator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="calculator">
    <input type="text" id="display" readonly>
    <div class="buttons">
      <button onclick="clearDisplay()">C</button>
      <button onclick="appendToDisplay(’/’)">/</button>
      <button onclick="appendToDisplay(’*’)">*</button>
      <button onclick="appendToDisplay(’-’)">-</button>
      <button onclick="appendToDisplay(’7’)">7</button>
      <button onclick="appendToDisplay(’8’)">8</button>
      <button onclick="appendToDisplay(’9’)">9</button>
      <button onclick="appendToDisplay(’+’)">+</button>
      <button onclick="appendToDisplay(’4’)">4</button>
      <button onclick="appendToDisplay(’5’)">5</button>
      <button onclick="appendToDisplay(’6’)">6</button>
      <button onclick="appendToDisplay(’1’)">1</button>
      <button onclick="appendToDisplay(’2’)">2</button>
      <button onclick="appendToDisplay(’3’)">3</button>
      <button onclick="appendToDisplay(’0’)">0</button>
```

```html
      <button onclick="appendToDisplay('.')">.</button>
      <button onclick="calculate()">=</button>
    </div>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

3.2 CSS (styles.css)

```css
body {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background-color: #f0f0f0;
  margin: 0;
  font-family: Arial, sans-serif;
}
.calculator {
  background-color: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0,0,0,0.2);
}
#display {
  width: 100%;
  height: 40px;
  margin-bottom: 10px;
  font-size: 20px;
  text-align: right;
  padding: 5px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
.buttons {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 5px;
}
button {
  padding: 15px;
  font-size: 18px;
  border: none;
  border-radius: 5px;
  background-color: #4CAF50;
  color: white;
  cursor: pointer;
}
button:hover {
  background-color: #45a049;
}
```

3.3 JavaScript (script.js)

```
let display = document.getElementById(’display’);

function appendToDisplay(value) {
  display.value += value;
}

function clearDisplay() {
  display.value = ’’;
}

function calculate() {
  try {
    display.value = eval(display.value);
  } catch (error) {
    display.value = ’Error’;
  }
}
```

4 AWS Infrastructure Setup

4.1 Step 1: Create a VPC

1. Log in to the AWS Management Console.
2. Navigate to the VPC service.
3. Create a VPC with CIDR block `10.0.0.0/16`.
4. Name it `CalculatorVPC`.
5. Enable DNS hostnames and DNS resolution.

4.2 Step 2: Create Subnets

1. Create two public subnets in different availability zones (e.g., `us-east-1a`, `us-east-1b`):
    - Subnet 1: `10.0.1.0/24` in `us-east-1a`.
    - Subnet 2: `10.0.2.0/24` in `us-east-1b`.
2. Associate an Internet Gateway with the VPC.
3. Update the route table to route `0.0.0.0/0` to the Internet Gateway.

4.3 Step 3: Launch EC2 Instances

1. Navigate to the EC2 service.
2. Launch two EC2 instances (e.g., `t2.micro`, Amazon Linux 2 AMI).
3. Place one instance in each subnet (`us-east-1a`, `us-east-1b`).
4. Assign a security group allowing HTTP (port 80) and SSH (port 22).
5. Use the following user data script to install a web server and deploy the calculator application:

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
cd /var/www/html
cat << 'EOF' > index.html
[Insert index.html content from above]
EOF
cat << 'EOF' > styles.css
[Insert styles.css content from above]
EOF
cat << 'EOF' > script.js
[Insert script.js content from above]
EOF
chown -R apache:apache /var/www/html
chmod -R 755 /var/www/html
```

4.4 Step 4: Create an Application Load Balancer

1. Navigate to the EC2 service and select `Load Balancers`.
2. Create an Application Load Balancer named `CalculatorALB`.
3. Configure it to listen on HTTP (port 80).
4. Select both subnets (`us-east-1a`, `us-east-1b`).
5. Create a target group named `CalculatorTG` with HTTP health checks on `/index.html`.
6. Register the two EC2 instances in the target group.

4.5 Step 5: Configure Security Groups

1. Create a security group for the ALB allowing inbound HTTP (port 80) from `0.0.0.0/0`.
2. Update the EC2 instances' security group to allow HTTP (port 80) from the ALB security group only.
3. Ensure SSH (port 22) is allowed from your IP for management.

4.6 Step 6: Test the Application

1. Obtain the ALB DNS name from the AWS Console.
2. Access the DNS name in a browser to verify the calculator application loads.
3. Test calculator functionality (e.g., addition, subtraction).
4. Stop one EC2 instance to verify the ALB routes traffic to the other instance.

5 Troubleshooting

- **Application not loading**: Check ALB health checks, EC2 instance status, and security group rules.
- **HTTP 504 errors**: Ensure EC2 instances are running and responding to health checks.
- **CSS/JavaScript not loading**: Verify file permissions and paths in `/var/www/html`.

6 Cleanup

To avoid charges, delete the following resources:

- EC2 instances
- Application Load Balancer
- Target group
- VPC and associated subnets, Internet Gateway, and route tables