**IST107 - Introduction to Internet Programming and Web Applications**

**Assignment #7**

***Assignment #7: Deploying a Contact List Application with Load Balancer on AWS and GitHub***

|  |  |  |
| --- | --- | --- |
| **Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Student No.: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |  |

You are required to complete the lab and record your answers. Rename the file using your first name and the lab number, e.g., ***washington7.1.docx***, and submit it through Canvas to receive marks for this lab.

**Introduction**

In this assignment, you’ll build and deploy a simple JavaScript application that manages a contact list. You’ll create and configure two EC2 instances and a load balancer on AWS to distribute traffic between these instances. Additionally, you’ll use GitHub to manage the project, with at least three branches for version control and development.

**Objectives**

1. Create and deploy a simple contact list application using JavaScript.
2. Set up two EC2 instances on AWS to run the application.
3. Configure a load balancer to distribute traffic across the instances.
4. Use GitHub for project management, including branching and repository setup.

**Instructions**

Please follow each part carefully, as each component is essential for successful deployment and testing.

**Part 1: Develop a JavaScript Application**

1. **Create the Application Code:**
   * Create a JavaScript file (e.g., contactList.js) that will manage an array of contact objects, with properties: name, phone, and email.
   * Initialize the contact list with three users, e.g.,
     + John Doe / 1234 567 890 / john.doe@example.com
     + Jane Smith / 2345 678 901 / jane.smith@example.com
     + Bob Johnson / 3456 789 012 / bob.johnson@example.com
   * Add functionality to:
   * Display the initial contact list.
   * Add a new contact with hard-coded data, e.g.,
     + Maisie Haley / 0913 531 3030 / risus.Quisque@urna.ca
     + Display the updated contact list.
   * Show the first and last contacts.
2. **Create the HTML File:**
   * Build a basic HTML file (e.g., index.html) that will include the contactList.js script. Ensure it displays the output within the console.
3. **Test Locally:**
   * Run your code in the browser console to confirm it works as expected, with all contacts displayed and updated accordingly.

**Part 2: Configure GitHub Repository**

1. **Set Up GitHub Repository:**
   * Create a GitHub repository for this project. Must be public repository.
   * Create at least three branches: main, development, and feature.
     + **main** – For the final deployment-ready version.
     + **development** – For ongoing integration and testing.
     + **feature** – For new features or changes before merging into development.
2. **Push Local Code to GitHub:**
   * Commit and push your code to the appropriate branches **(main, development, feature).**
   * Submit the GitHub repository URL with the assignment.

**Part 3: Deploy on AWS**

1. **Create Two EC2 Instances:**
   * Launch two EC2 instances using Amazon Linux 2023 AMI, t2.micro instance type, and 35 GiB storage (gp3).
   * Name the instances "WebServer1" and "WebServer2."
   * Configure a security group allowing HTTP (Port 80), HTTPS (Port 443), and SSH (Port 22).
2. **Install Apache and Deploy Application:**
   * On each instance, install and run the Apache HTTP server
   * Deploy your HTML and JavaScript files to each instance’s web root directory.
   * Edit the HTML on each instance to differentiate the content slightly (e.g., "Web Server 1" and "Web Server 2").

**Part 4: Set Up Load Balancer**

1. **Create an Application Load Balancer:**
   * In AWS, create an Application Load Balancer and set it to be internet-facing with HTTP on port 80.
2. **Configure a Target Group:**
   * Create a target group, adding both EC2 instances with HTTP protocol and port 80.
   * Keep the default health check settings and attach the target group to the load balancer.
3. **Finalize the Setup:**
   * Verify that the load balancer is functioning correctly.

**Part 5: Testing and Verification**

1. **Access Load Balancer DNS:**
   * Locate the DNS name of the load balancer in the EC2 dashboard.
   * In a browser, access this DNS name and confirm that traffic is distributed between "Web Server 1" and "Web Server 2" upon refreshing.
2. **Screenshots:**
   * **Provide screenshots of:**
     + Each EC2 instance with the application running.

Here your screenshot

Here your screenshot

* + - The DNS load balancer, showing traffic distribution between both servers.

Here your screenshot

Here your screenshot

* + **Include a screenshot of your GitHub repository URL and branches.**

Here your GitHub repository link

**Submission Instructions**

1. **Compile Your Work:**
   * Organize all screenshots and relevant details in a Word document.
   * Rename the file with your name and assignment number, e.g., **Assignment7\_YourName.docx**.
2. **Submit via Canvas:**

You are required to complete the lab and record your answers. Rename the file using your first name and the lab number, e.g., ***washington7.docx***, and submit it through Canvas to receive marks for this lab.