

## **IST105 - Introduction to Programming Assignment #4**

### **Assignment #4 - Building an Interactive Web Application with Conditional Logic, Version Control, and Deployment on AWS with Load Balancing**

#### **Important Note:**

- Assignments with code that contains errors and does not run correctly will receive a grade of zero.
- Please ensure that your code is fully functional and free of syntax or runtime errors before submitting. It is recommended that you thoroughly test your code in Python IDLE and fix any issues that arise.
- Include comments where necessary to explain your logic and observations.

#### **Prerequisite:**

1. **AWS EC2 Setup:**
  - Ensure you have access to AWS and know how to create EC2 instances.
  - Familiarize yourself with creating and configuring a load balancer.
2. **Software Installation:**
  - Familiarize yourself with installing and configuring Python, PHP, and Apache on an EC2 instance.
3. **Version Control with Git/GitHub:**
  - Understand how to create and manage repositories, branches, and commits on GitHub.
4. **Basic Programming Knowledge:**
  - Understand Python syntax for basic operations and the use of conditional statements.

#### **Objective:**

The goal of this assignment is to build an interactive web application that collects user input, performs conditional checks and calculations using Python, and dynamically displays results using PHP. You will also deploy the application on AWS using two EC2 instances and a load balancer configured with the Least Connection method. This assignment will deepen your understanding of web development, user input handling, backend processing with conditional logic, version control, and load balancing.

#### **Assignment Tasks:**

## 1. Set Up AWS Infrastructure:

- Launch **two EC2 instances** using the Amazon Linux AMI.
- Configure security groups to allow HTTP (port 80) and SSH (port 22) access.
- **Create a load balancer and configure it to use Weighted random - new.**
- Note the public IP addresses of your instances and the DNS name of the load balancer.

## 2. Install Required Software:

- On both EC2 instances:
  - Install Apache web server, PHP, and Python.
  - Start and enable the Apache service.

## 3. Create the User Input Form:

- Write a PHP script named form.php that includes a form where users can input three numerical values (a, b, and c).
- The form should have the following fields:
  - Input for a (numeric)
  - Input for b (numeric)
  - Input for c (numeric)
  - A submit button labeled "Calculate".

## 4. Create the Python Script:

- Write a Python script named calculate.py that:
  - Retrieves the values of a, b, and c from the user input.
  - Performs conditional checks and calculations:
    - Check if a, b, and c are numeric. If not, return an error message.
    - If a is less than 1, display a message indicating that the input is too small.
    - If b is equal to 0, indicate that it will not affect the result.
    - If c is negative, provide a specific error message.
    - If c is greater than or equal to 0, compute  $c^3$ .
    - If the result of  $c^3$  is greater than 1000:
      - Multiply the square root of  $c^3$  by 10.
      - Otherwise, divide the square root by a.
    - Add b to the final result.
  - Generate output formatted as HTML to display the results on a web page.

## 5. Create the PHP Script to Process Input:

- Write a PHP script named process.php that:
  - Receives the user input from form.php.
  - Calls calculate.py and passes the input values.
  - Displays the results returned from the Python script or any error messages.

## 6. Set Up Version Control with Git/GitHub:

- Create a **public GitHub repository** named IST105-Assignment4.
- Create **three branches**:
  - main: Contains the final, working version of the code.
  - development: Used for testing and integrating features.
  - feature1: Used for developing the initial version of the application.
- Commit and push all files (form.php, calculate.py, and process.php) to all three branches.

## 7. Deploy the Application:

- Upload the files from the main branch to the **/var/www/html/** directory on both EC2 instances.
- Ensure the Python script has the correct permissions to be executed:

```
sudo chmod +x /var/www/html/calculate.py
```

## 8. Test the Setup:

- Open your web browser and navigate to the load balancer's DNS name (e.g., <http://<load-balancer-dns>/form.php>) to see the user input form.
- Submit values to see the calculations and output displayed on the results page.

## Submission:

### 1. GitHub Repository:

- Provide the **public link** to your GitHub repository.

<https://github.com/DeboraSerra/Assignment4-IST105>

- Ensure the repository contains:
  - form.php
  - calculate.py
  - process.php
  - All files must be present in the main, development, and feature1 branches.

### 2. Screenshots:

- A screenshot of the web page with the Python script running, **showing the public IP addresses** of both EC2 instances.



## Welcome to server 1

Input a value for a

Input a value for b

Input a value for c

Calculate



## Welcome to server 2

Input a value for a

Input a value for b

Input a value for c

Calculate

# Welcome to server 2

Input a value for a

Input a value for b

Input a value for c

Calculate

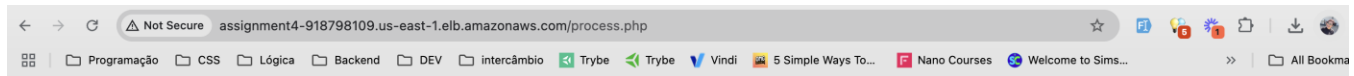
# Welcome to server 1

Input a value for a

Input a value for b

Input a value for c

Calculate

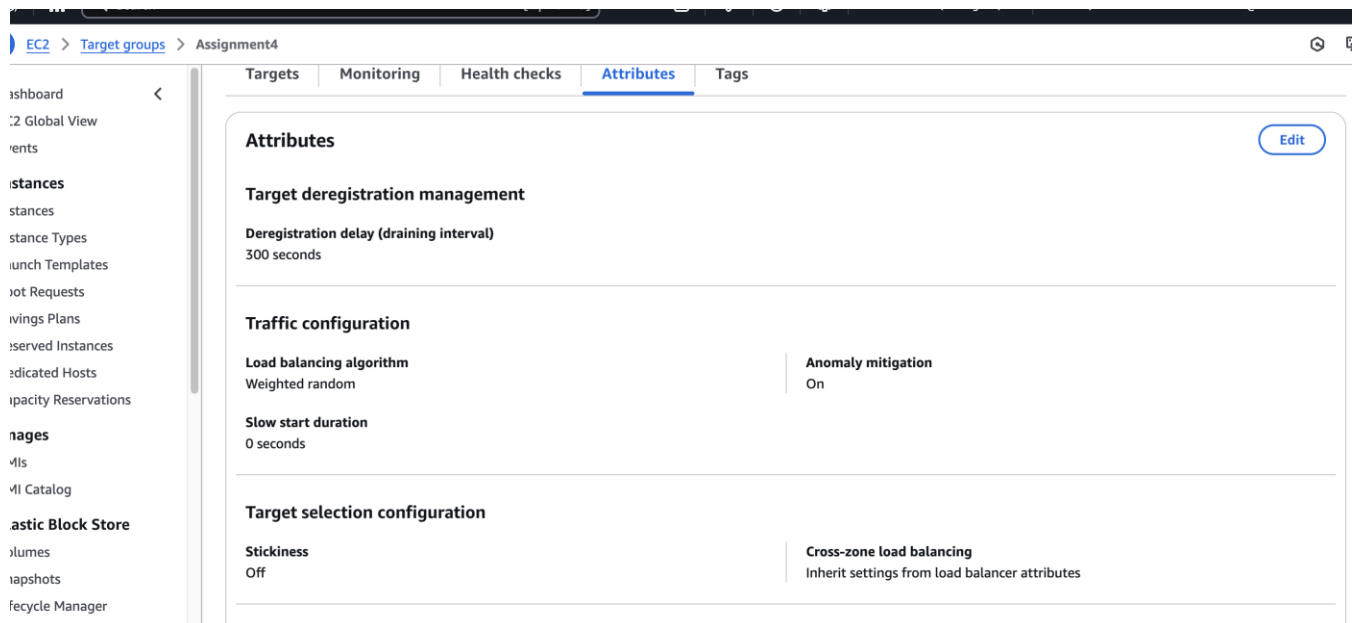


## Assignment 4

### Error

All values must be integers.

- A screenshot of the load balancer configuration, verifying the use of the **Weighted random - new**.



### 3. File:

- Upload this file named **Assignment4\_FirstNameLastName.doc** to Canvas containing:
  - The GitHub repository link.
  - All screenshots.

**Tips:**

- Ensure all paths and permissions are set correctly to avoid access issues.
- Test your Python and PHP scripts locally before deploying them on the EC2 instances.
- Double-check the public IP addresses and load balancer DNS name.
- Review the Apache and Python integration settings to ensure smooth operation.
- Use meaningful commit messages and branch names in your GitHub repository.
- Apply the knowledge and techniques from previous labs in your script. Your implementation should reflect the skills and concepts covered in our previous labs.