1 Overview

In this MP, we are going to bring up a load balancer to balance requests between 2 servers (Amazon EC2 instances).

2 Requirements

You need to use the terminal or command prompt on your computer. You also need to ssh into AWS instances. For Mac and Linux users, ssh is already available in the terminal. For Windows users, you can install PuTTY.

3 Procedure

Step 1:

- Bring up 2 EC2 instances. We suggest using the latest Amazon Linux AMI as the OS, and the smallest EC2 instance to make sure your EC2 instances cost as low as possible, or are free.

- Make sure that you are able to connect to the 2 EC2 instances using your browser or your terminal using ssh.

- You need to write a simple server application which stores and retrieves a seed value (default 0) that will run on a specific port on both of these EC2 instances and handle the following two HTTP REST requests:

HTTP POST: "/" with JSON body {"num": 100} where 100 can be any integer. This should update the seed value with the number.

HTTP GET: "/" -> Returns the integer seed value in string format. The response body for the above case will be: "100"

You can use any of your preferred languages or systems to implement this server however we will only be able to assist you with python using flask server. You need to install the required dependencies (install *pip* and then *flask* along with its required dependencies if you are using python) in both of the EC2 instances and then start the server. In order to access the server publicly and make it available for the load balancer, you need to set an inbound rule for the EC2 instances for your server port.

Step 2:

You are going to bring up an AWS Elastic Load Balancer, which will allow to distribute requests between the 2 instances you have just brought up. You might want to refer to the following document to get started.

https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-getting-started.html

Using the tutorial, connect your load balancer to the previously brought up 2 EC2 instances. Make sure to map the correct ports on the load balancer.

Make sure that you do not use the default security configuration when setting up the load balancer. When creating a custom security configuration, allow traffic to be routed from anywhere and using the HTTP protocol. Unless our autograder can access your system on the internet, you won't pass this assignment.

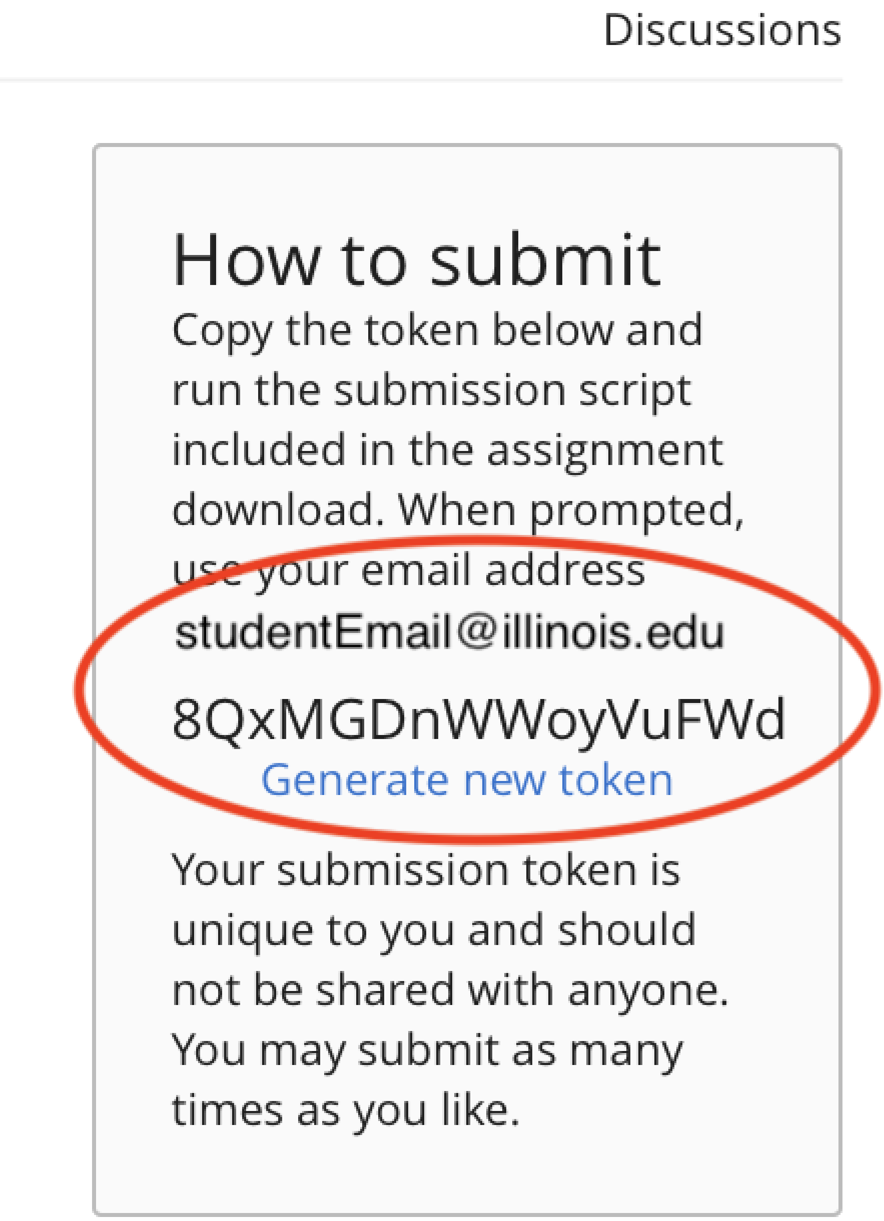
To make the load balancer work with the server code, you will have to edit the load balancer's health check ping path from "/index.html" to "/".

To test to see whether your load balancer is working correctly, load the DNS address given in the description of the AWS load balancer in your favorite browser, and you should see a number.

If you encounter any other problems, use Google Search, AWS documentation, or Piazza.

Step 3:

Edit the variable payload in test.py (see the attached file below) and add relevant details. You can get your "secret information from this page, look for an information like the following at the top of this assignment page:



Discussions

How to submit

Copy the token below and run the submission script included in the assignment download. When prompted, use your email address

studentEmail@illinois.edu

8QxMGDnWWoyVuFWd

Your submission token is unique to you and should not be shared with anyone. You may submit as many times as you like.

Run test.py on your local machine (or any of the two EC2 instances). You will receive the autograder's feedback for your submission as a response to the test.py script and your grade on coursera will be updated.

Step 4:

Make sure to shutdown your EC2 instances, otherwise Amazon will keep charging you!

MP Files: