**Load Test Setup Procedure**

一張含有 文字, 名片 的圖片

自動產生的描述

In the load balancing measurement setup, we aim to simulate the situation of multiple concurrent users accessing the same network resource as a webpage to the exact location. Then we will install a load balance strategy such as HAproxy, EC2 load balancer, and Nginx onto the frontend server. Based on the technologies and methods that we choose; the frontend server will forward the request to one of the agent servers to distribute the user load. We expect that using this can achieve a higher performance than a single node server. In addition, we will compare the technologies we mentioned above and decide which is the best and justify the reason.

1. **Install a web server to accept user request in backend**

We can build a simple website with a web page that accepts user requests and returns the result using Spring.

一張含有 文字 的圖片

自動產生的描述

Then we build the web container file

一張含有 文字 的圖片

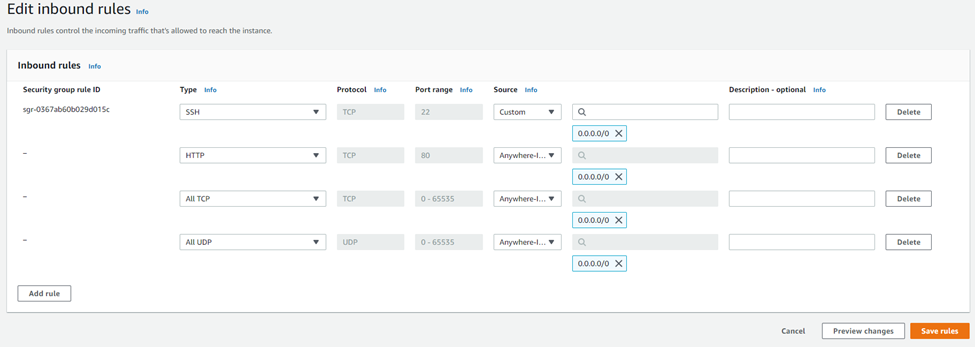
自動產生的描述

The program source code please refer to GitHub location:

<https://github.com/CityUGroup3/CityUGroup3/blob/main/Load_test_procedure/loadbalancetestweb-0.0.1-SNAPSHOT.war>

Create a **t2.micro** instance on Amazon cloud and choose **Ubuntu Server 18.04** as the AMI image.

Also, it is required to update the inbound rules for accepting internet connection from outside.



Then, upload the web container file (.war file) to the previous built t2.micro instance home directory. Alternative, you can run “**wget https://github.com/CityUGroup3/CityUGroup3/raw/main/Load\_test\_procedure/loadbalancetestweb-0.0.1-SNAPSHOT.war --no-check-certificate**” to download war file directly.

一張含有 文字 的圖片

自動產生的描述

To start up the server on one of the backend servers, we will require to install java JDK.

Run "**sudo apt-get update**" first if it is the first time running the EC2 instant.

Then run “**sudo apt install openjdk-11-jdk**”

一張含有 文字 的圖片

自動產生的描述

After that, we can start the web server by running “**sudo java -jar loadbalancetestweb-0.0.1-SNAPSHOT.war**”

一張含有 文字 的圖片

自動產生的描述

Now the server should be up and running. We can access the testing website with the EC2 instance public IP with port 80

Repeat steps in this section to set up 20 EC2 instance for backend web servers and they will be used in load balancing.

1. **Setup frontend server to accept requests from the user and perform load balance strategy**

In this step, we will be required to set up the environment for running the load balancing solution. Each of them needs to configure differently. Therefore, the following step will be separated into three subsections.

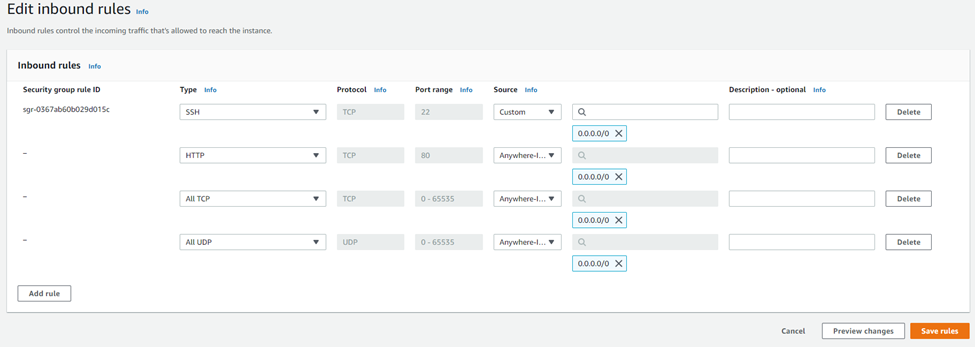
**2.1 HAproxy load balancer setup procedure.**

Create a **t2.micro** instance on Amazon cloud with **Ubuntu Server 18.04** as the AMI image, but this time we will use it to install different types of load balancing strategies.

一張含有 文字 的圖片

自動產生的描述

It is also required to update the inbound rules for accepting internet connection from outside.



Remote into the EC2 instance and run "**sudo apt-get update**" if it is the first startup

Then run “**sudo apt install haproxy**” to install HAproxy.

一張含有 文字 的圖片

自動產生的描述

Haproxy should run automatically, to check the service status, run "**sudo systemctl status haproxy**"

一張含有 文字 的圖片

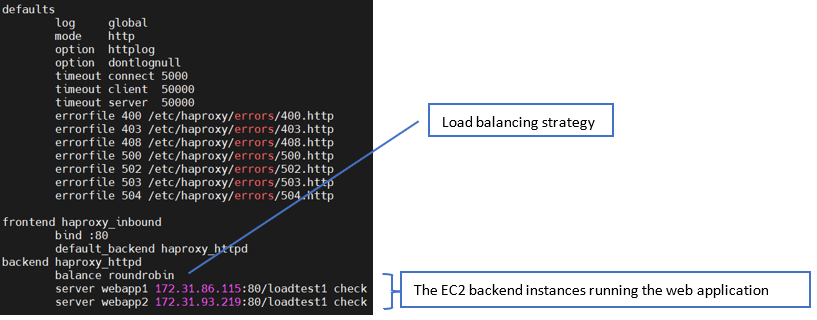
自動產生的描述

After that, we will need to configure our load balancer. Go to directory **/etc/haproxy/** and find **haproxy.cfg**

一張含有 文字 的圖片

自動產生的描述

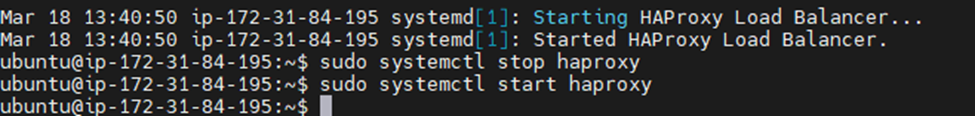
Add the following lines to setup the load balance strategy and the **private IP** of backend servers that we just set in step 1.



Config file (with lease connection as load balancing method) can refer to GitHub location:

<https://github.com/CityUGroup3/CityUGroup3/blob/main/Load_test_procedure/config/haproxy/haproxy.cfg>

To make the new setting to take effect, we need to restart the HAproxy service by running "**sudo systemctl stop haproxy**" then "**sudo systemctl start haproxy**"



Then the load balancer should be working correctly. Access the load balancer **public IP** address to test the result.

一張含有 文字 的圖片

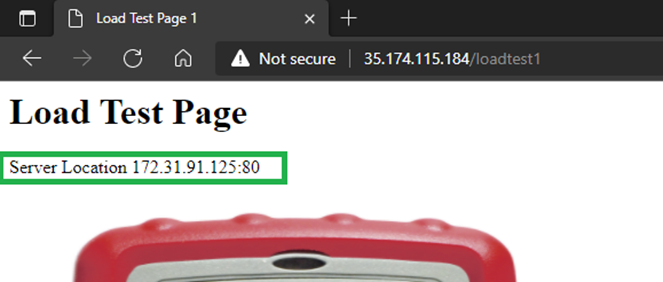
自動產生的描述

You can check the result of the webpage

一張含有 文字, 行動電話, 手機, 陳列 的圖片

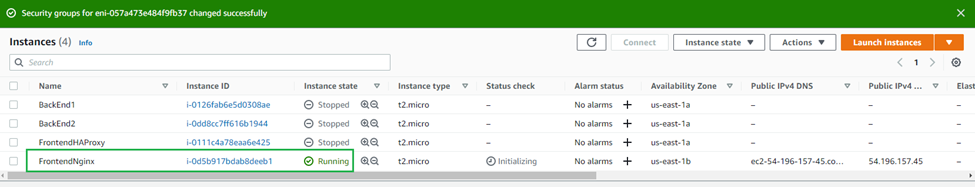
自動產生的描述

Since it is round robin, you can refresh the webpage and see each backend server taking turns serving the page.

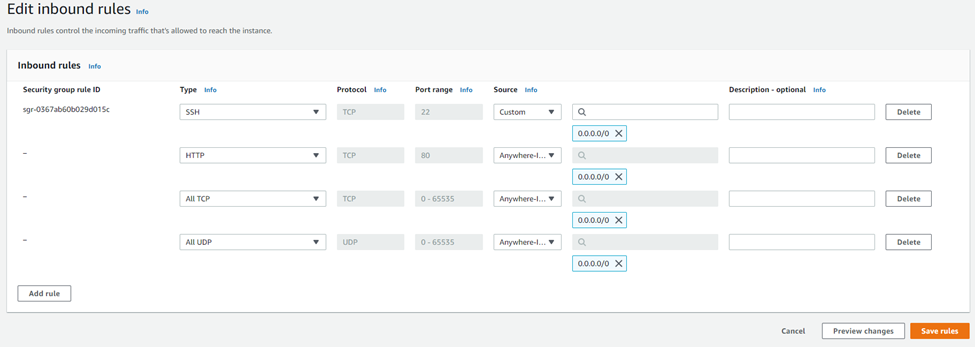


**2.2 Nginx load balancer setup procedure**

Again, we c Create a **t2.micro** instance on Amazon cloud with **Ubuntu Server 18.04** as the AMI image to install our load balancer with Nginx.



It is also required to update the inbound rules for accepting internet connection from outside.



Run "**sudo apt-get update**" if it is the first startup. Then run “**sudo apt install nginx**” to install nginx.

一張含有 文字 的圖片

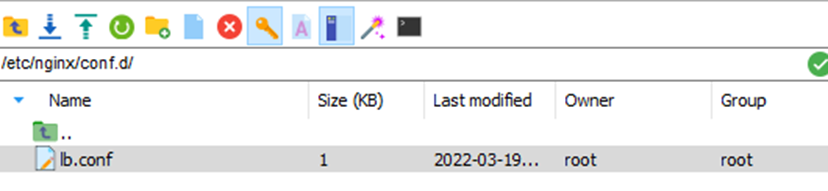
自動產生的描述

Check service’s running status with command “**sudo systemctl status nginx**”

一張含有 文字, 監視器, 螢幕, 螢幕擷取畫面 的圖片

自動產生的描述

Go the directory location **/etc/nginx/conf.d/** and find **lb.conf**



Upload the following configuration file with load balancing details (Ref to GitHub). It is required to update the IP with your EC2 instance web servers’ private IP.

<https://github.com/CityUGroup3/CityUGroup3/blob/main/Load_test_procedure/config/nginx/lb.conf>

一張含有 文字 的圖片

自動產生的描述

Run "**sudo nginx -s reload**" or "**sudo systemctl stop nginx**" follow with "**sudo systemctl start nginx**" to rstart the Nginx server to let the new setting takes effect.

一張含有 文字 的圖片

自動產生的描述

You can now browse webpage with Nginx public IP with **port 8080** and refresh the webpage to check if it is success.

**2.3 AWS EC2 load balancer setup procedure.**

Unlike Nginx and HAProxy, EC2 load balance in a build-in service in Amazon AWS. This means it does not require a dedicated server to hold the balancing solution.

一張含有 文字 的圖片

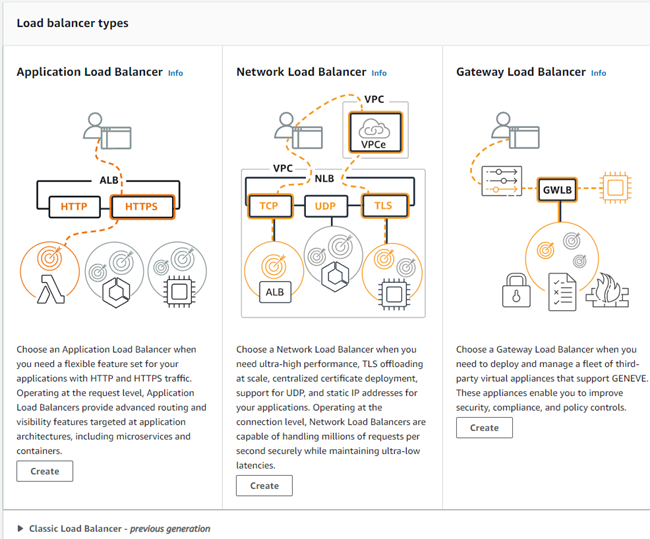
自動產生的描述

We can locate the load balancing service by typing "EC2 Load Balancer" on AWS website.

一張含有 文字 的圖片

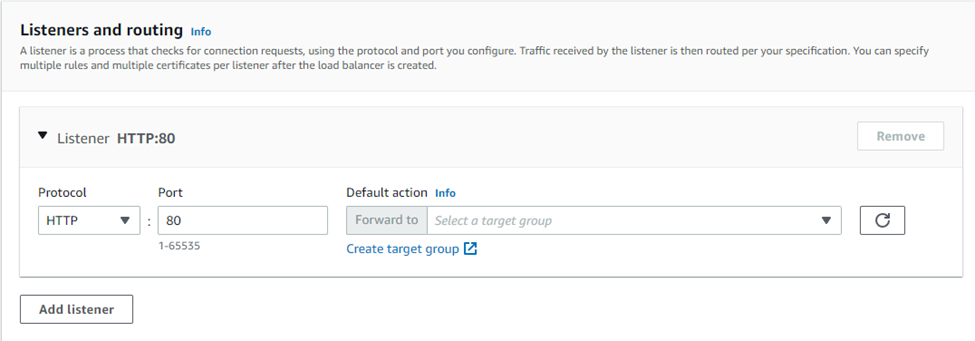
自動產生的描述

Click Create Load Balancer

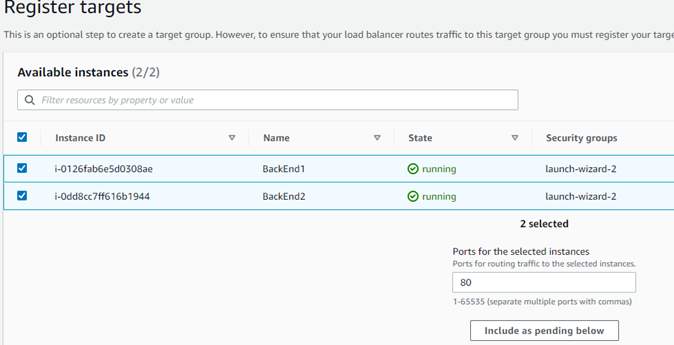


Choose the type of load balance we want to create, this time we use **application load balancer**

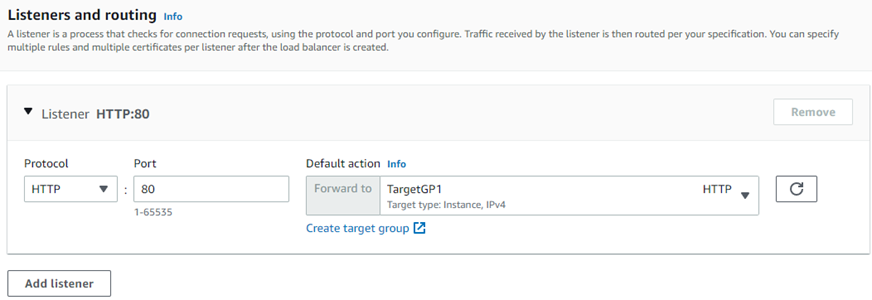
The setup should be straightforward. You just need to type the name and select the parameter it requires.

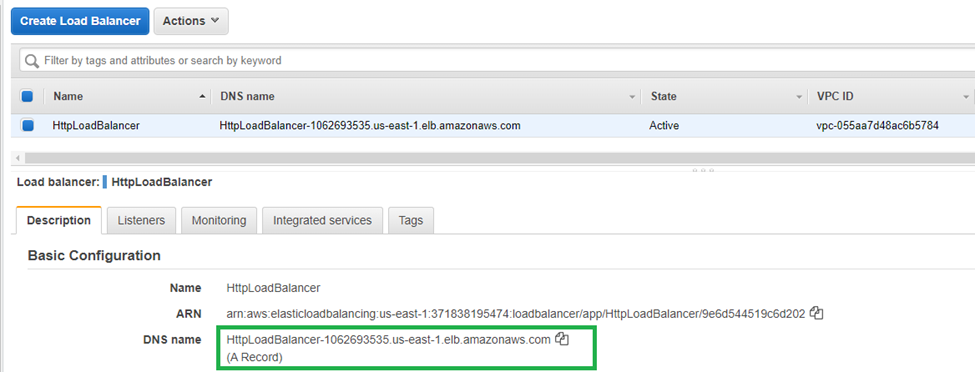


One thing we need to be careful is adding the target group, which assigns the instance as backend servers for load balancing.



Assign EC2 instances into the target group. Then create the load balancer.





Copy the load balancer “DNS name” and access the site.

一張含有 文字, 監視器 的圖片

自動產生的描述

As you can see, it directed to our EC2 instance from the target group we assigned. This means the load balance was created successfully.