**Configuring NGINX Load balancer and HA proxy in AWS using Terraform**

**NGINX:**

NGINX is a free, open-source web server that can also be used as a load balancer, reverse proxy, and more. It's known for its high performance and low resource utilization.

**Nginx load balancer:**

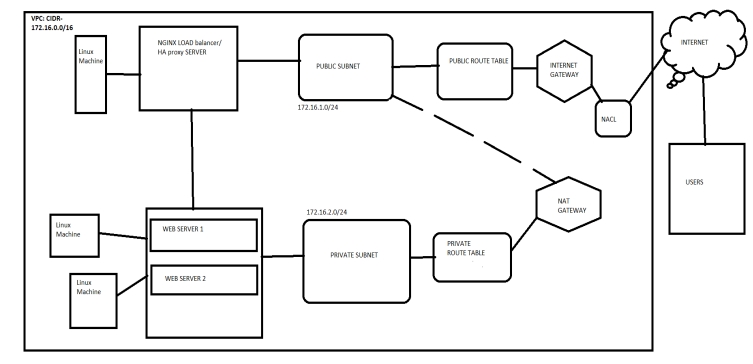
In load balancing setup, NGINX distributes incoming traffic among multiple backend servers. This is done to improve scalability, availability, and performance by spreading the requests evenly.

**HA(high availability) proxy**:

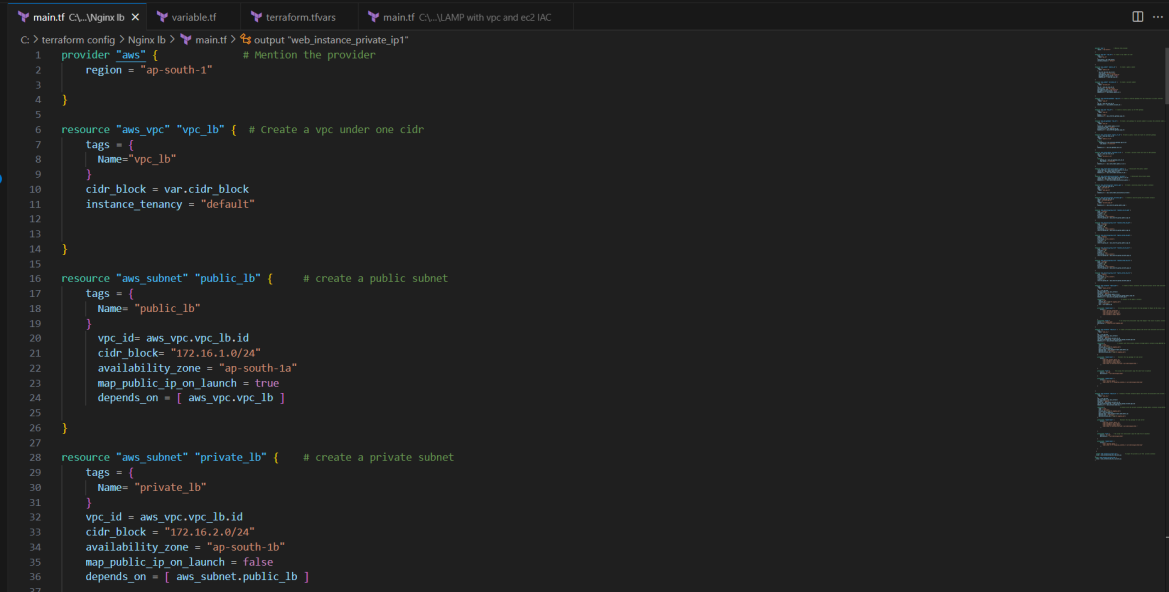
* + HAProxy is specifically designed as a high-performance load balancer, proxy server and it do health checks.
  + It is often used in environments where the primary focus is load balancing and high availability for HTTP/HTTPS, TCP, and SSL/TLS traffic.
  + HAProxy is often favored in enterprise environments or more complex architectures, as it offers very fine-grained control over load balancing algorithms and health checks.

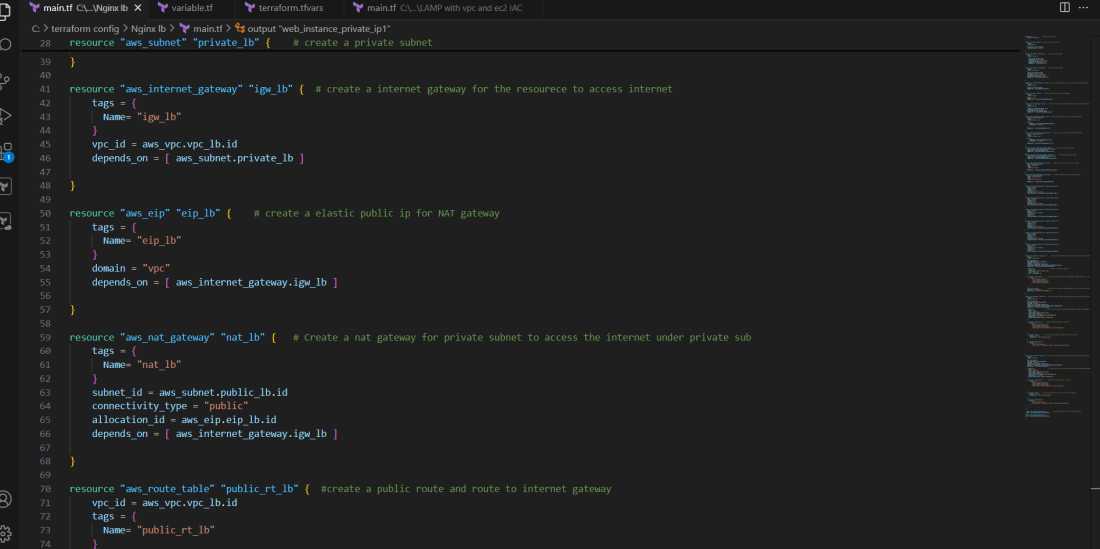
|  |  |  |
| --- | --- | --- |
| Feature | NGINX Load Balancing | HAProxy Load Balancing |
| Configuration Structure | Uses upstream block for backend servers and proxy\_pass. | Uses frontend and backend sections for traffic routing. |
| Load Balancing Algorithms | Round-robin (default), least connections, IP hash, etc. | Round-robin (default), leastconn, source (and many more). |
| Health Checks | No built-in health checks; requires external modules. | Built-in health checks for backend servers using check. |
| Session Persistence | Can use ip\_hash for session persistence. | Uses stick-table for session persistence, or source. |
| Use Case | Web server + reverse proxy + load balancer. | Specialized **high-performance** load balancer. |

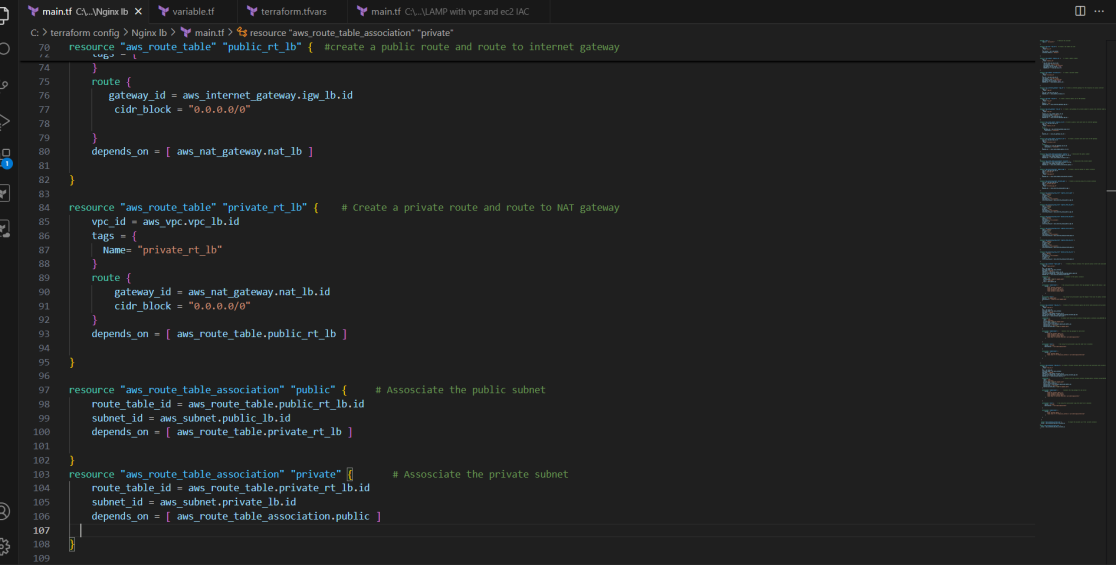
**Architecture:**

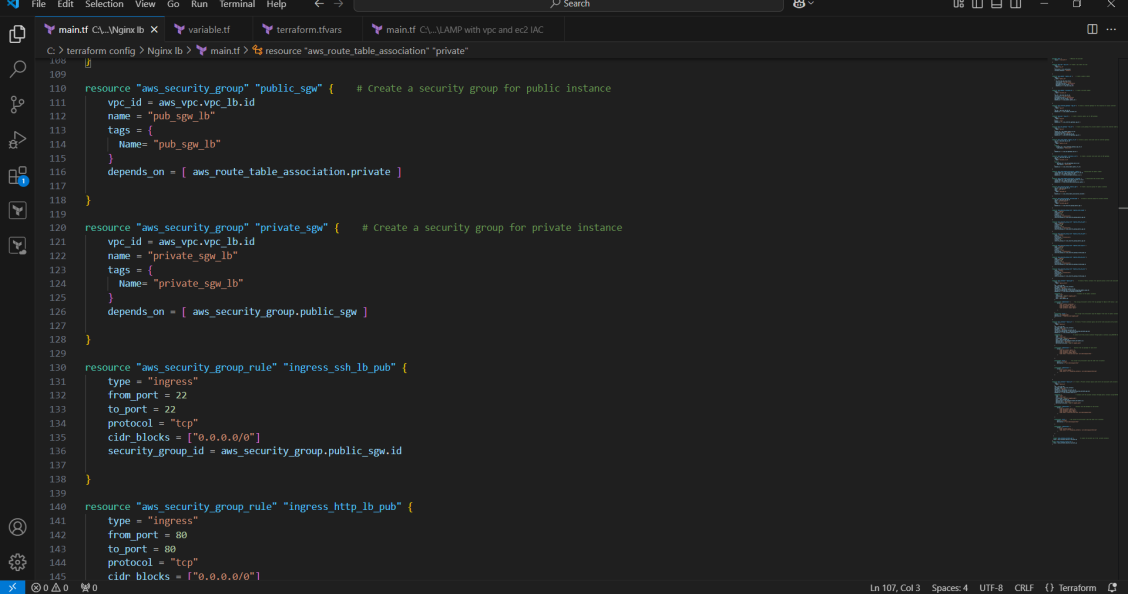


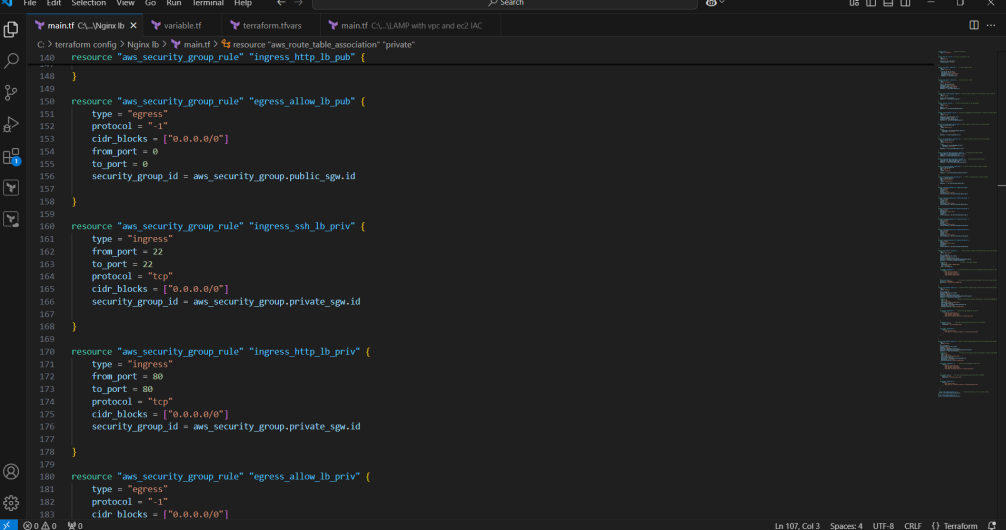
Terraform code for configuration:

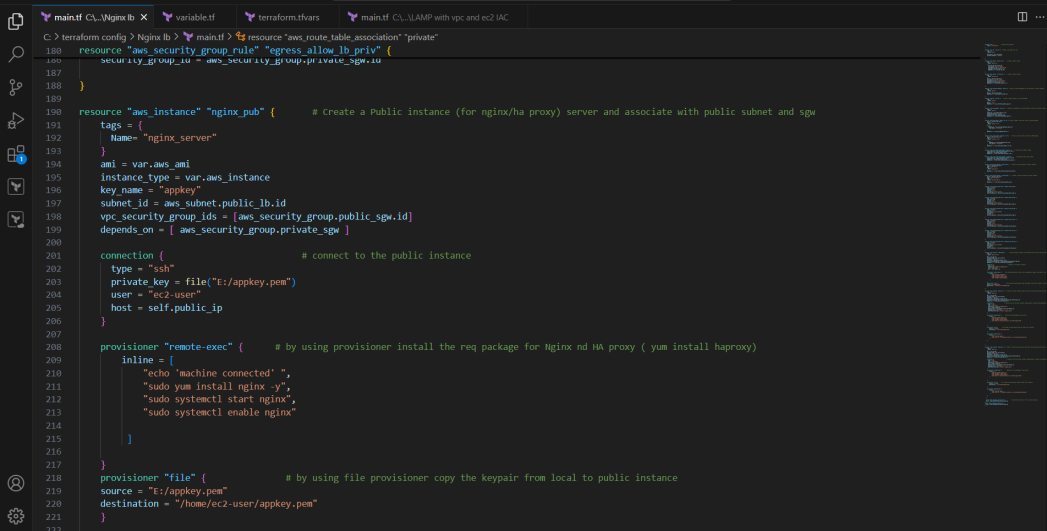


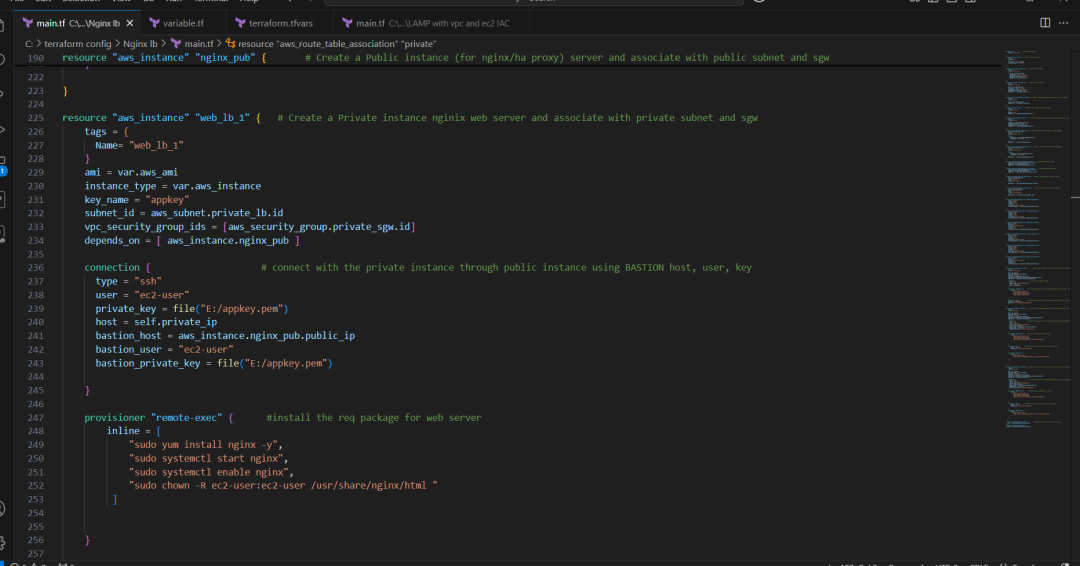


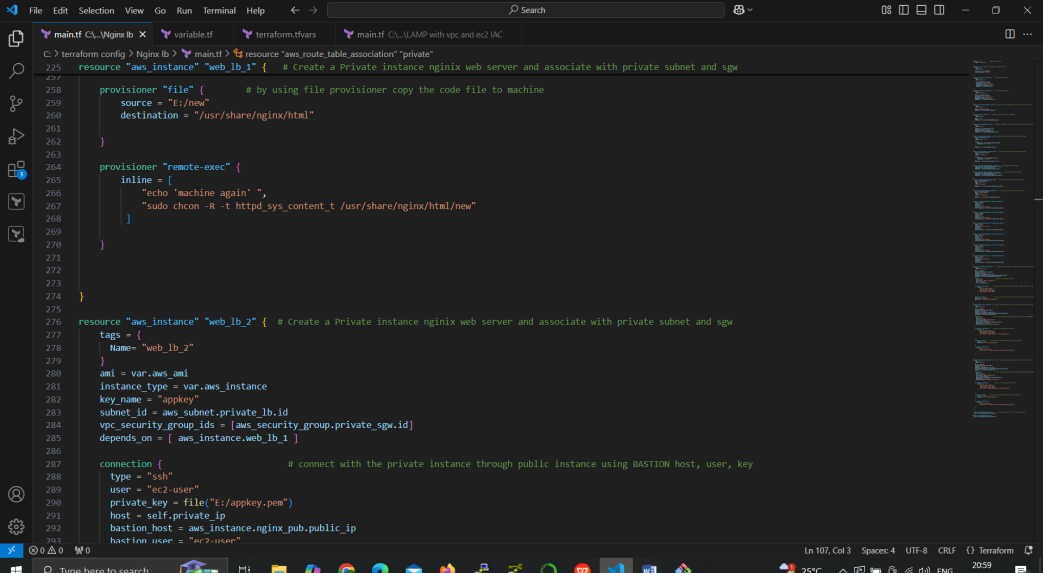


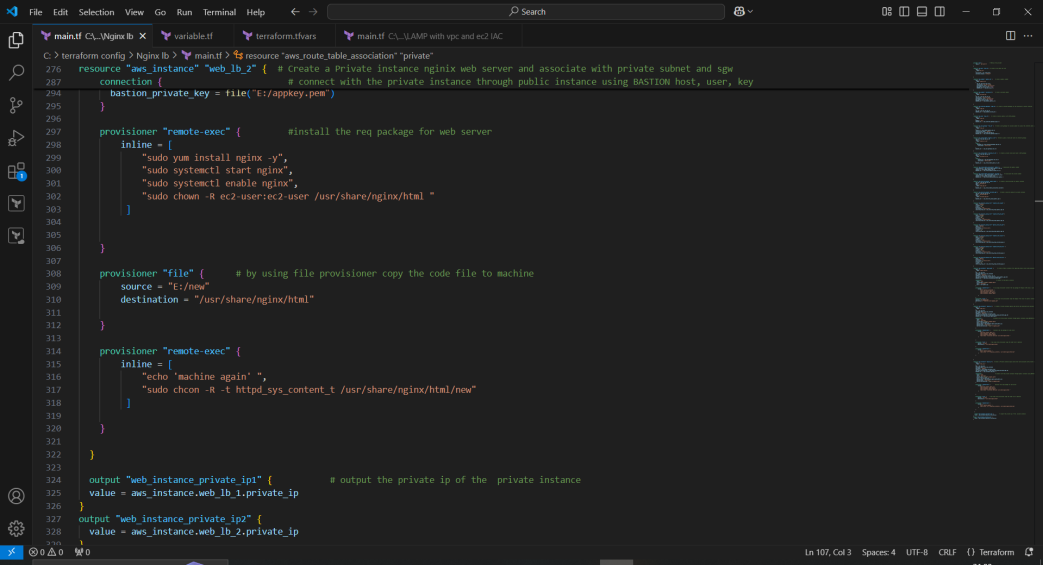


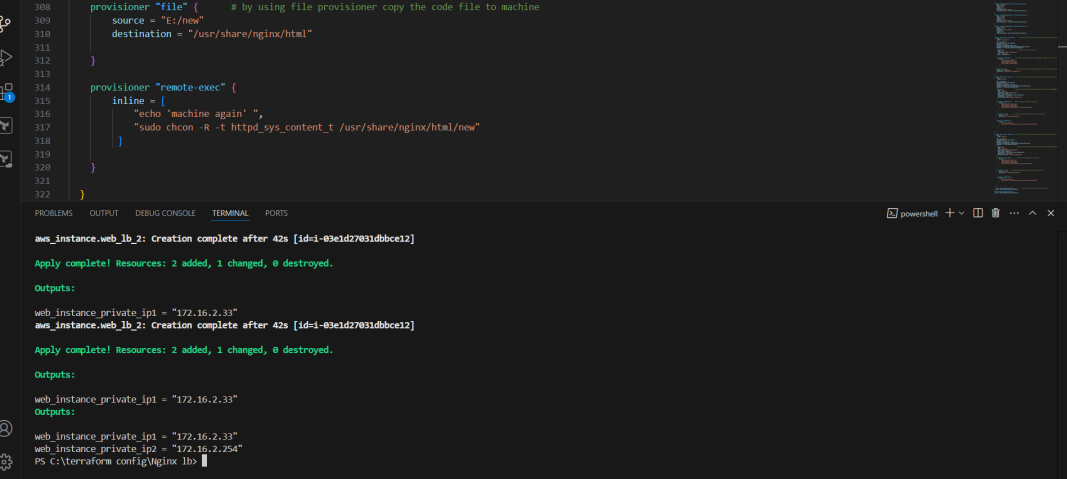


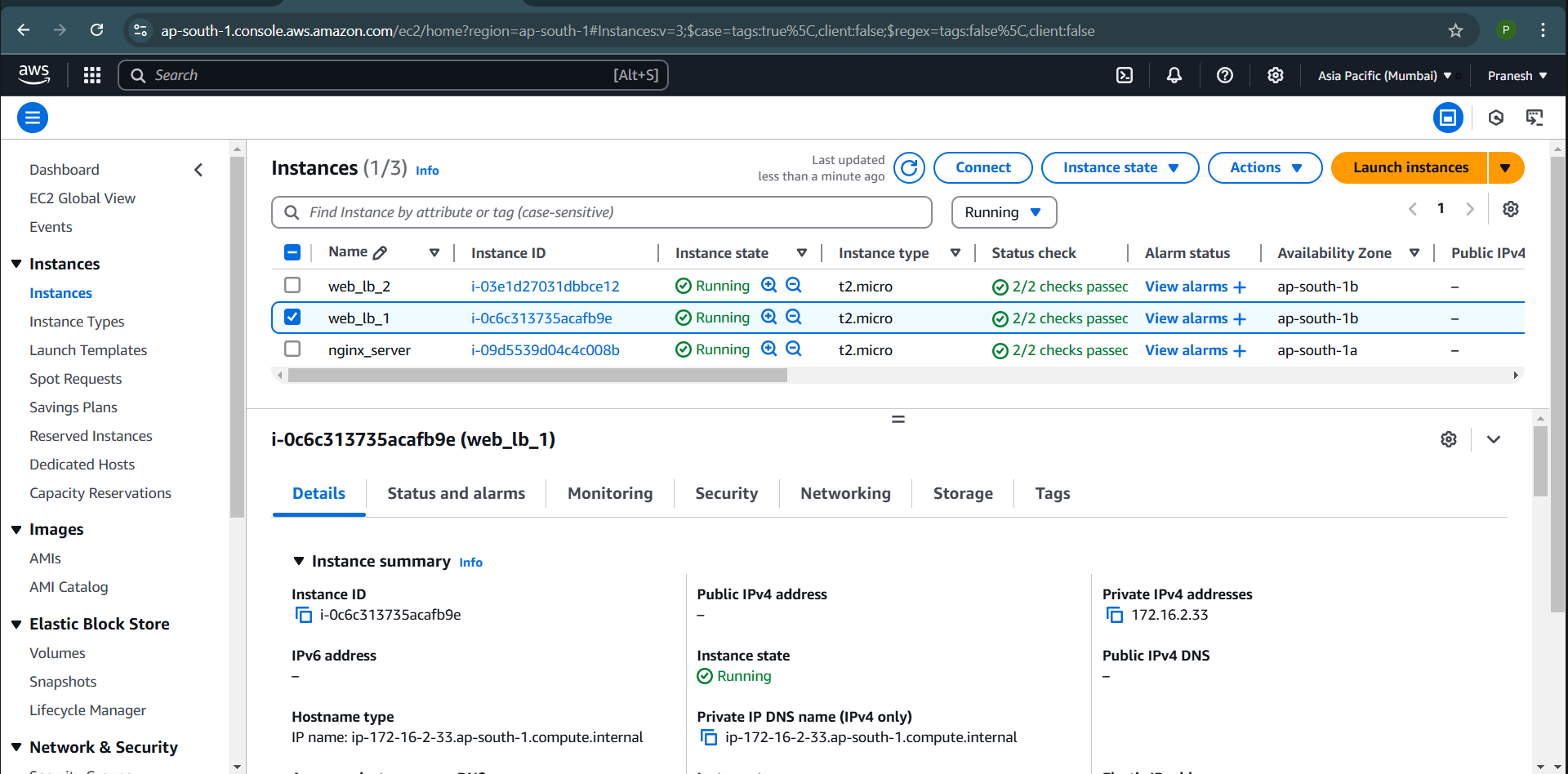


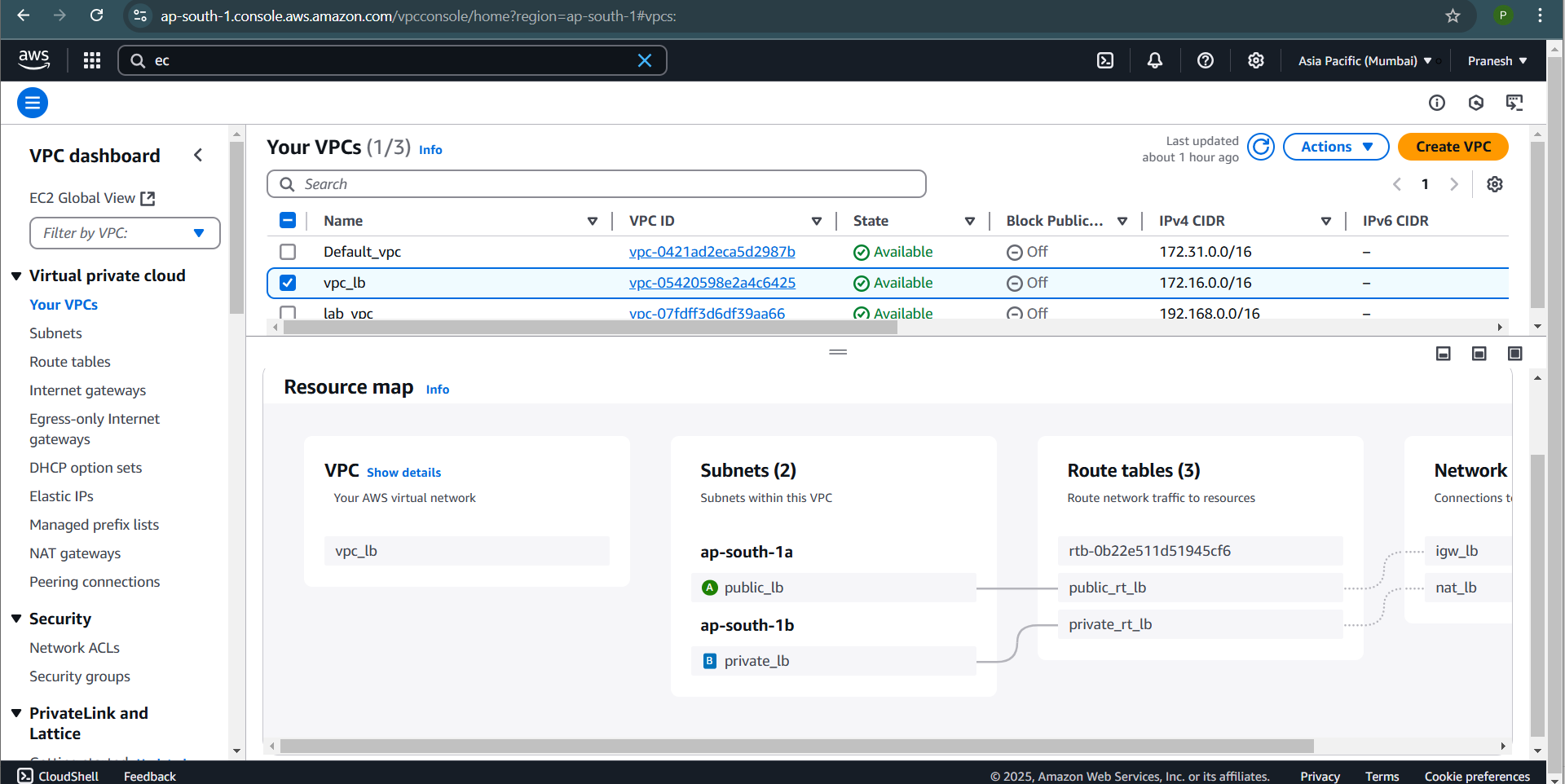




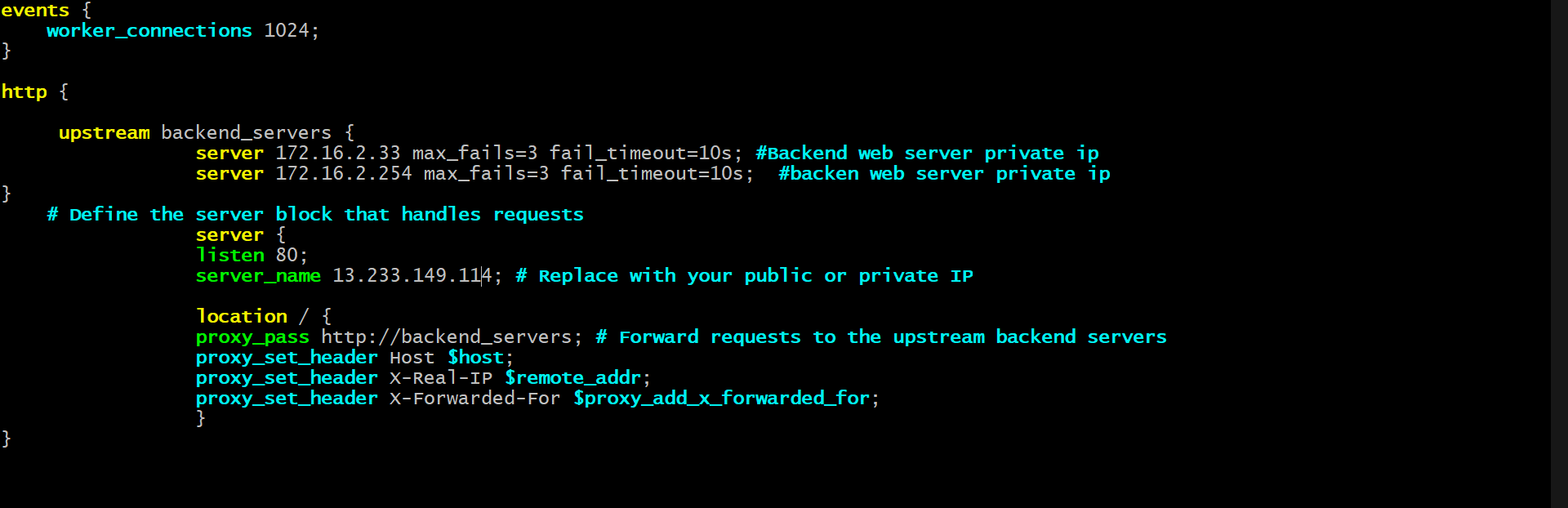








Now open the Nginx server and go to /etc/nginx/nginx.conf and edit the conf for the server to act as Load balancer and add the Backend server (web server) IP:

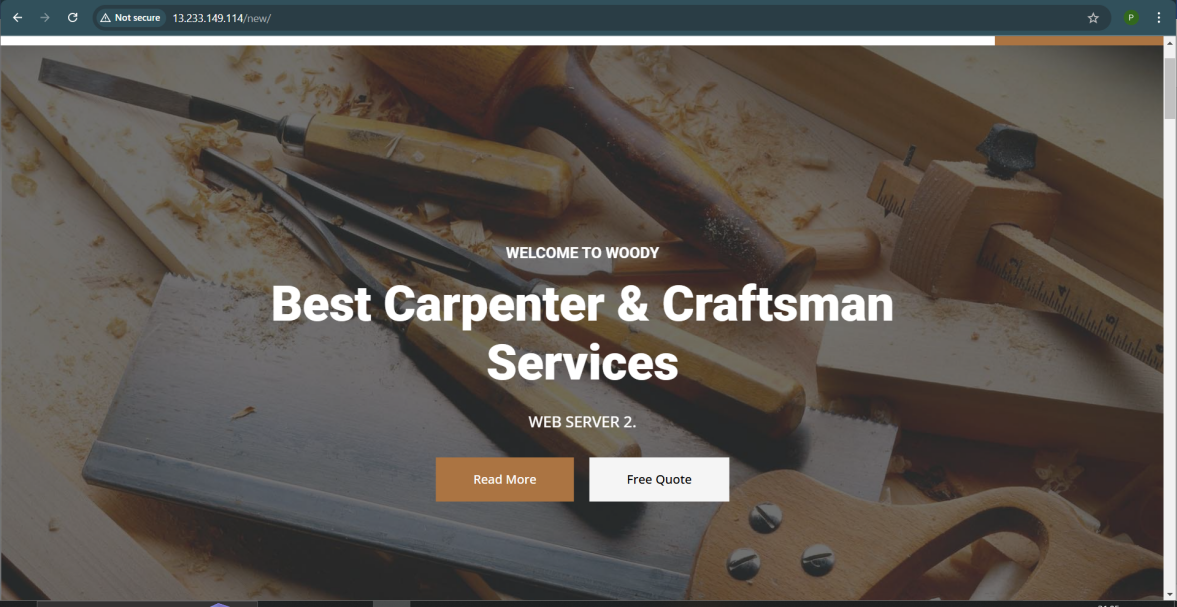


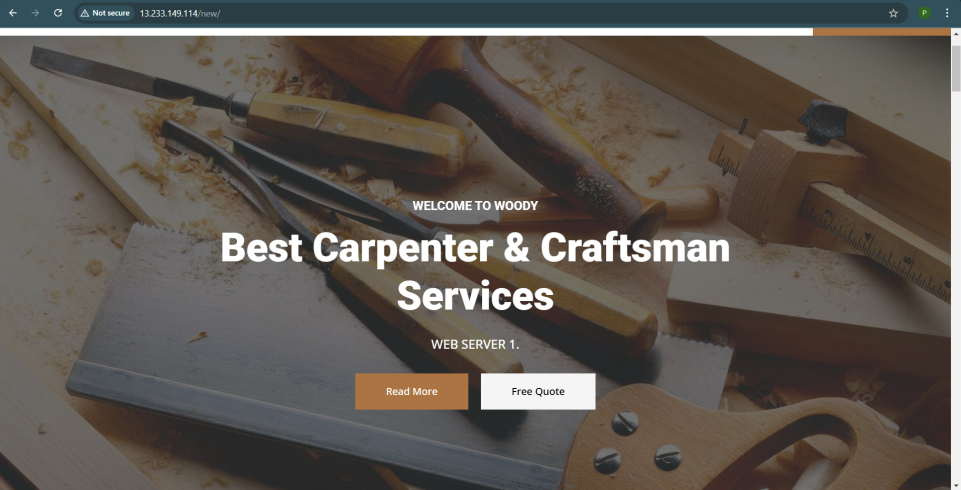
Test the conf with nginx -t

Reload the nginx

Now check the load balancer is distributing the traffic evenly by

Giving IP of LB server with path in search:

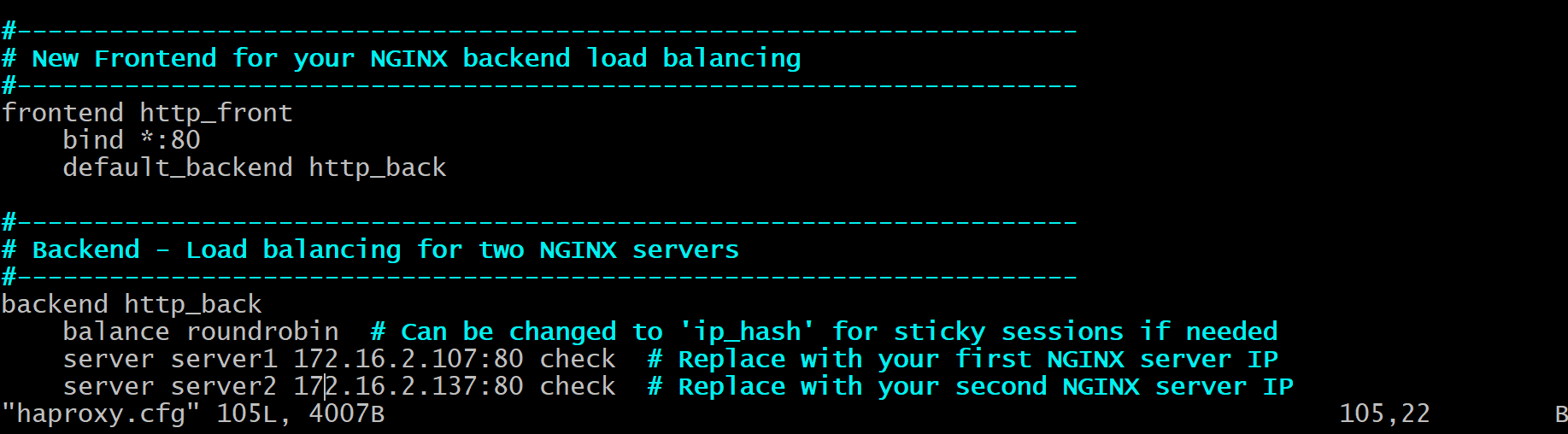




**HA PROXY:**

SAME FOR HA PROXY BUT INSTALL WITH PACKAGE **yum install haproxy -y** and START THE SERVICE.

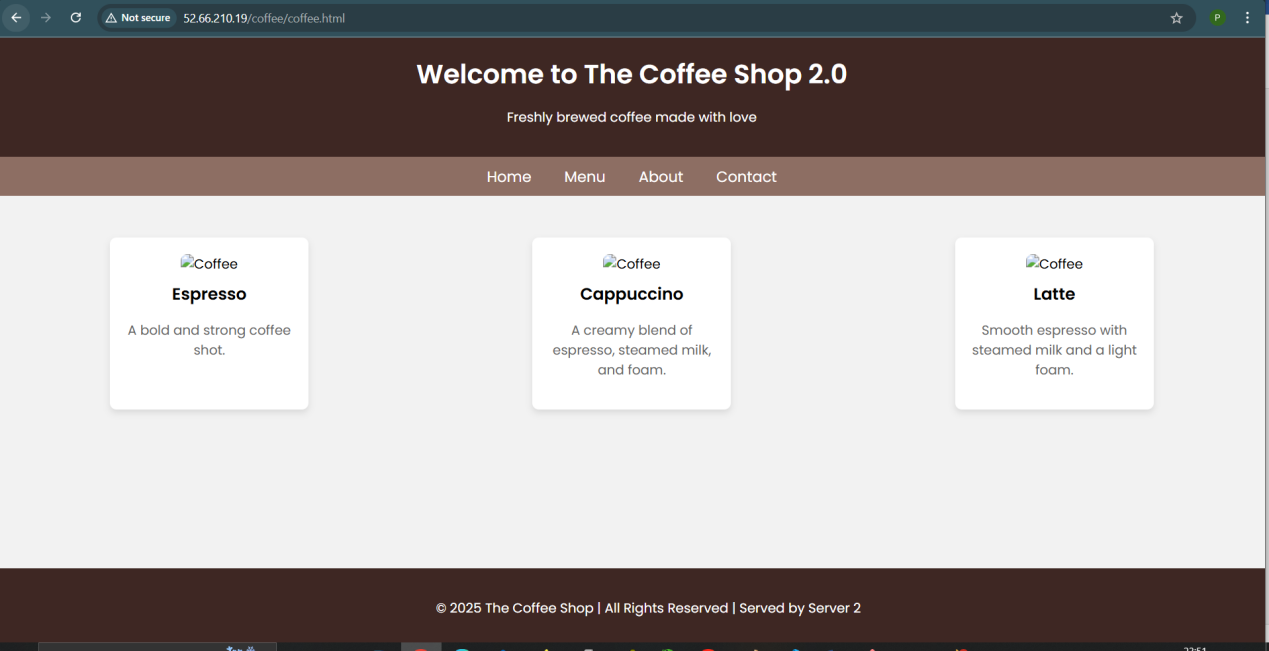
AND EDIT THE ONF FILE TO MAKE THE SERVER AS HAPROXY IN PATH /etc/haproxy/haproxy.cfg



Test the conf file by running cmd sudo haproxy -f /etc/haproxy/haproxy.cfg-c

Once valid run systemctl reload haproxy.

Now check in the search engine by running public ip of haproxy server with file path.



You can see for every hit it changes the load to both machine one after another.