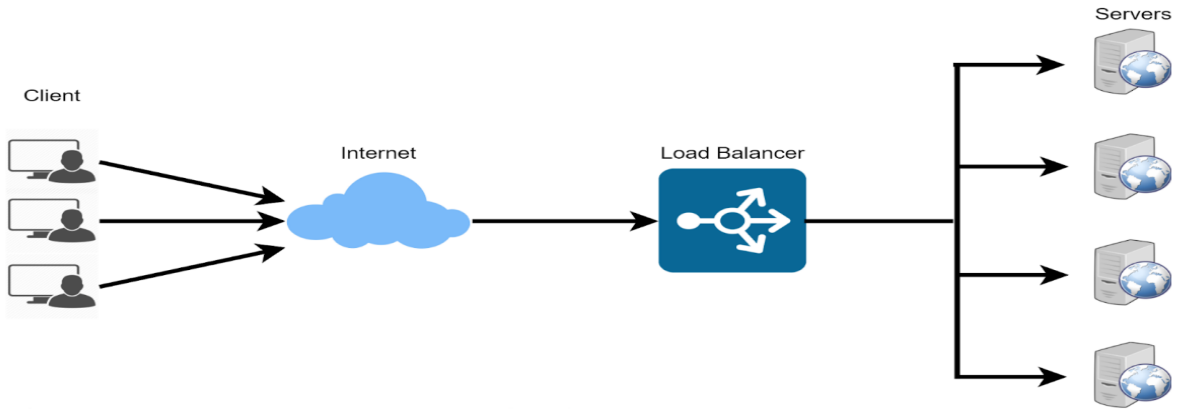
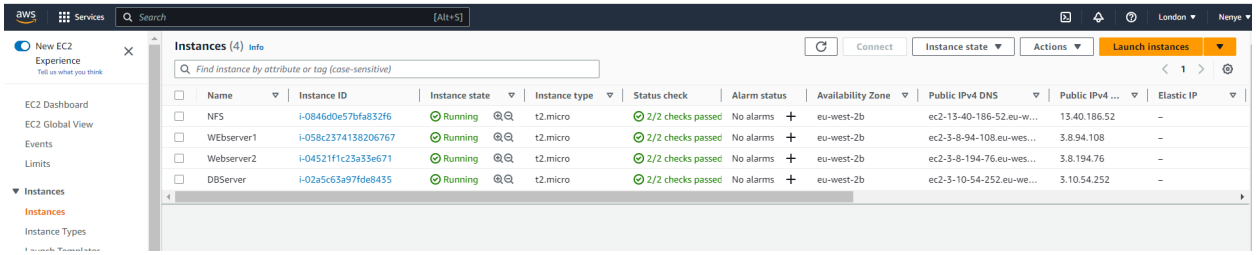
PROJECT 8: LOAD BALANCER SOLUTION WITH APACHE

Any web server's performance has typically been impacted by web traffic, which causes the server to become overloaded and respond slowly. There must be an ideal solution for the overload issue that each web server experiences. Load-balancing techniques are thus used to get around this. It serves as a transmitter between the client's requests and the web servers by spreading the incoming requests among all of the cluster's nodes to maximise speed and resource use and ensure scalable, affordable, and cost-effective operation. A simple basic illustration can be found below.



To further optimise our Tooling website from Project 7, I will deploy 1 Database server running on Ubuntu, 1 NFS server, and 2 Web servers on RHEL. These web servers will be load balanced so that we may visit our website from a single point of access using a single public IP address or its DNS.

Launch the EC2 Instance



Configure Apache As A Load Balancer

Please refer to Project 7 for the database installation. Next is to install Apache Load Balancer on the server and set it up to direct traffic to both web servers from Load Balancer.

```
ubuntu@ip-172-31-46-217:~$ sudo apt update
```

```
ubuntu@ip-172-31-46-217:~$ sudo apt install mysql-server -y
```

```
ubuntu@ip-172-31-46-217:/$ sudo apt-get install libxml2-dev
```

Enable the following modules:

sudo a2enmod rewrite

sudo a2enmod proxy

sudo a2enmod proxy_balancer

sudo a2enmod proxy_http

sudo a2enmod headers

sudo a2enmod lbmethod_bytraffic

And for each enabled module, the result should appear as enabling module in the below screenshot.

```
ubuntu@ip-172-31-46-217:/$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
ubuntu@ip-172-31-46-217:/$ sudo a2enmod proxy
Enabling module proxy.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

It is ideal to ideal to restart apache2 and confirm that the service is running with

```
ubuntu@ip-172-31-46-217:/$ sudo systemctl restart apache2
ubuntu@ip-172-31-46-217:/$ sudo systemctl status apache2
```

```
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-06-17 15:16:49 UTC; 12s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 8180 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 8184 (apache2)
      Tasks: 55 (limit: 1141)
     Memory: 5.0M
     CGroup: /system.slice/apache2.service
             ├─8184 /usr/sbin/apache2 -k start
             ├─8185 /usr/sbin/apache2 -k start
             └─8186 /usr/sbin/apache2 -k start

Jun 17 15:16:48 ip-172-31-46-217 systemd[1]: Starting The Apache HTTP Server...
Jun 17 15:16:49 ip-172-31-46-217 systemd[1]: Started The Apache HTTP Server.
```

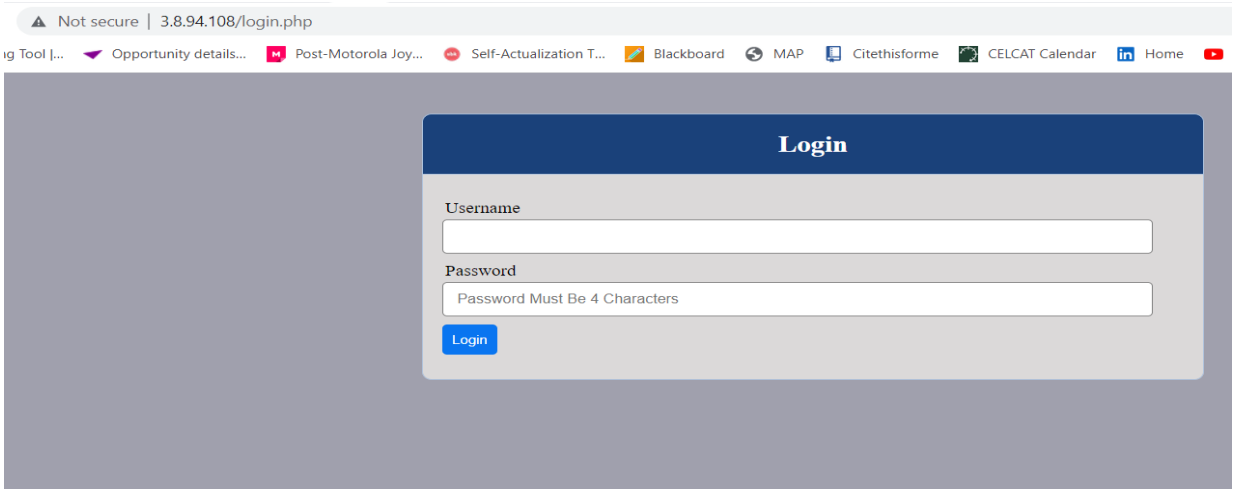
Edit and configure load balancing by adding a balancing method that will distribute the incoming load between the Web Servers according to the current traffic load. Webserver 1 and 2 Private IP addresses will be used for this configuration. Ensure to restart the Apache2 server with the sudo systemctl restart apache2 command.

```
ubuntu@ip-172-31-46-217:/$ sudo vi /etc/apache2/sites-available/000-default.conf
```

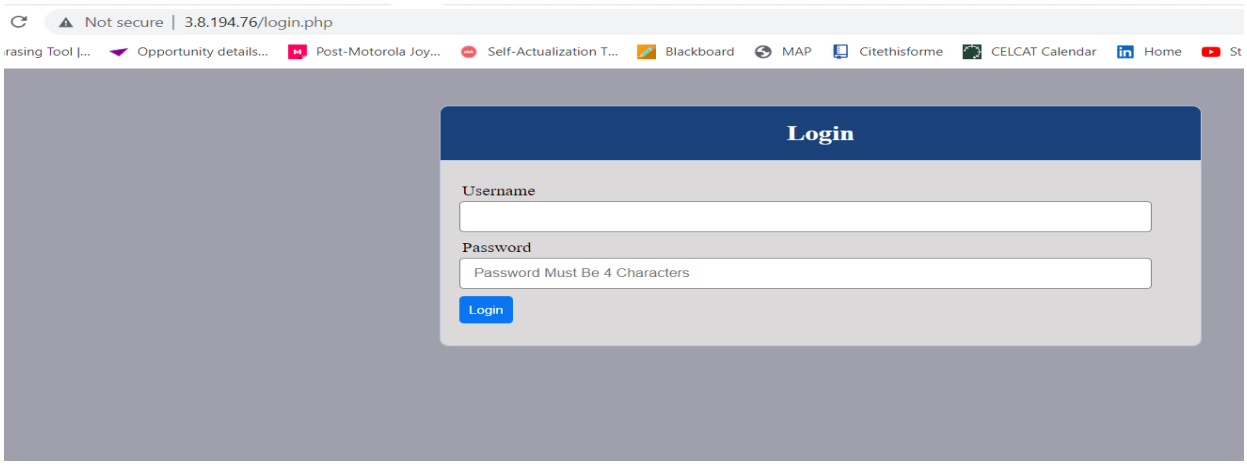
```
    <Proxy "balancer://mycluster">
           BalancerMember http://172.31.39.100:80 loadfactor=5 timeout=1
           BalancerMember http://172.31.34.99:80 loadfactor=5 timeout=1
           ProxySet lbmethod=bytraffic
           # ProxySet lbmethod=byrequests
    </Proxy>

    ProxyPreserveHost On
    ProxyPass / balancer://mycluster/
    ProxyPassReverse / balancer://mycluster/
```

To verify that our configuration works from the browser use the public IP or DNS in this format: http://<Load-Balancer-Public-IP-Address-or-Public-DNS-Name>/index.php and this should resolve in the screenshots below.

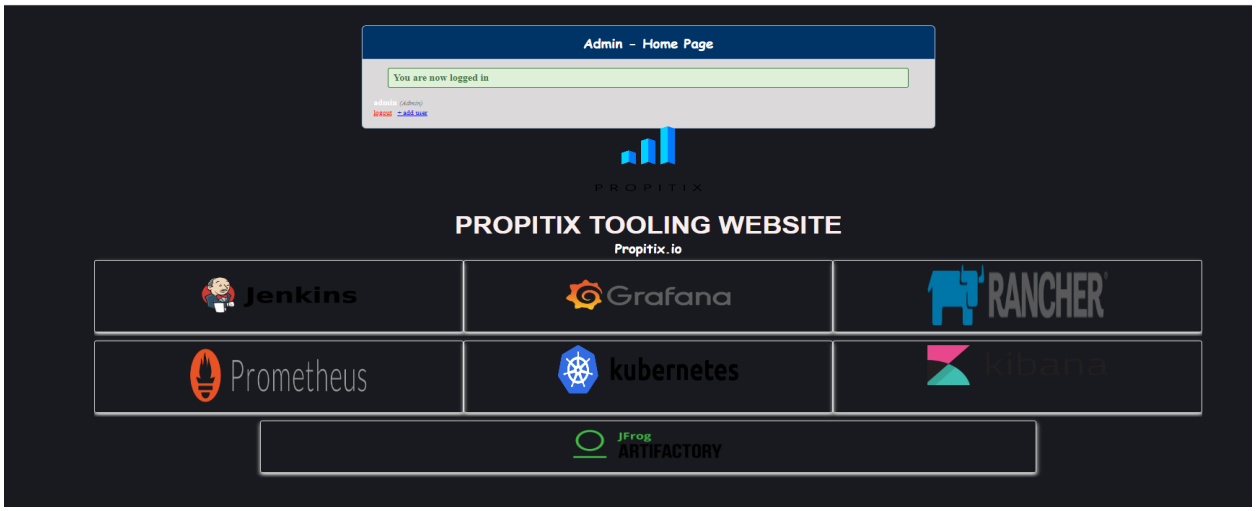
Web server 1



Web server 2

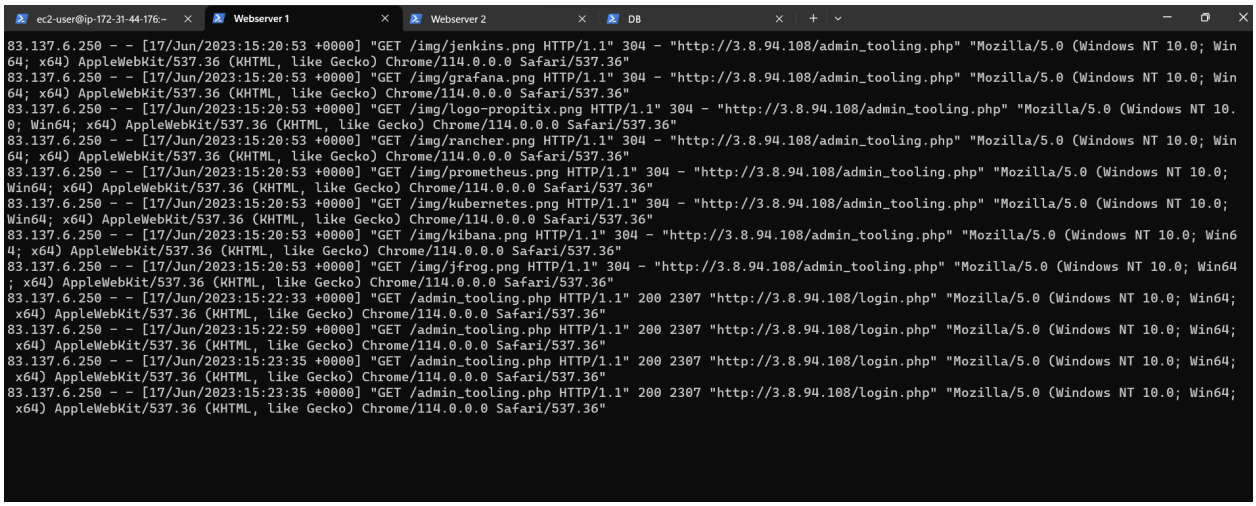


Tooling website displayed for both web servers

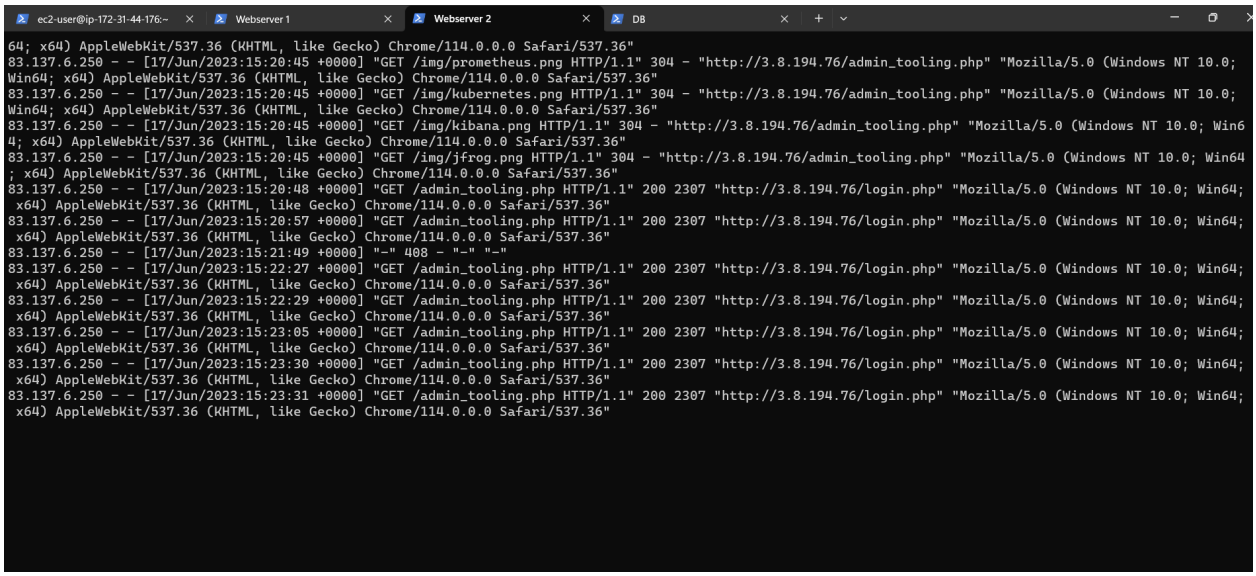


To confirm if /var/log/httpd/ is mounted from the Web Servers to the NFS server in Project 7 use the *sudo umount -f /var/log/httpd* command. Then open two ssh consoles for both Web Servers and run the *sudo tail -f /var/log/httpd/access_log* command. I refreshed both browsers several times to ensure that both servers HTTP GET requests from the load balancer in the server log file. The number of requests to each server will be approximately the same since we set the load factor to the same value for both servers meaning that traffic will be distributed evenly between them.

Web server 1



Web server 2



Configure Local DNS Resolution

Configuring local domain name resolution allows for us to use the DNS rather than an IP address. These changes can be made in the use /etc/hosts file on the load balancer by using *sudo vi /etc/hosts.* The WebServer1-Private-IP-Address and WebServer2-Private-IP-Address will be saved as: *172.31.39.100 Web1* *172.31.34.99 Web2*

Then update the Load balancer config file with Web1 and Web2 instead of IP addresses and then save the changes.

```
ubuntu@ip-172-31-46-217:/$ sudo vi /etc/apache2/sites-available/000-default.conf
```

```
    <Proxy "balancer://mycluster">
            BalancerMember http://Web1:80 loadfactor=5 timeout=1
            BalancerMember http://Web2:80 loadfactor=5 timeout=1
            ProxySet lbmethod=bytraffic
            # ProxySet lbmethod=byrequests
    </Proxy>
```

Making a curl request to both Web Servers from the load balancer locally will now resolve to an HTML-formatted version. Please note that this is only an internal configuration and will only apply to the load balancer server.

```
</html>ubuntu@ip-172-31-46-217:/$ curl http://Web1


<!DOCTYPE html>

<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="tooling_stylesheets.css">
   <script src="script.js"></script>
    <title> PROPITIX TOOLING</title>
</head>


<body>



<div class="header">

       </div>
       <div class="content">
              <!-- notification message -->
                            <!-- logged in user information -->
              <div class="profile_info">
              <!--    <img src="images/user_profile.png"  > -->

                     <div>
                                                  </div>
              </div>
       </div>
```

```
ubuntu@ip-172-31-46-217:/$ curl http://Web2


<!DOCTYPE html>

<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="tooling_stylesheets.css">
   <script src="script.js"></script>
    <title> PROPITIX TOOLING</title>
</head>


<body>



<div class="header">

       </div>
       <div class="content">
              <!-- notification message -->
                            <!-- logged in user information -->
              <div class="profile_info">
              <!--    <img src="images/user_profile.png"  > -->

                     <div>
                                                  </div>
              </div>
       </div>
```

CONCLUSION

By splitting traffic into separate requests and selecting the server that will fulfil each request, the load balancer installation for our Tooling website will assist maintain high site performance and ensure that our web servers are always able to handle requests.

REFERENCES

Bourke, T., 2001. Server load balancing. " O'Reilly Media, Inc.".

Evans, M. (2021) What is a Load Balancer and its Types? Available at: https://www.cloud4u.com/blog/what-is-a-load-balancer-and-its-types/ (Accessed: 17 June 2023).

Ibrahim, I.M., Ameen, S.Y., Yasin, H.M., Omar, N., Kak, S.F., Rashid, Z.N., Salih, A.A., Salim, N.O. and Ahmed, D.M., 2021. Web server performance improvement using dynamic load balancing techniques: A review. system, 19, p.21.