LOAD BALANCER SOLUTION WITH APACHE

After completing Devops Tooling Website Solution (project 7) you might wonder how a user will be accessing each of the webservers using 3 different IP addreses or 3 different DNS names. You might also wonder what is the point of having 3 different servers doing the same thing.

In our set up in Project-7 we had 3 Web Servers and each of them had its own public IP address and public DNS name. A client has to access them by using different URLs, which is not a nice user experience to remember addresses/names of even 3 server, let alone millions of Google servers.

In order to hide all this complexity and to have a single point of access with a single public IP address/name, a Load Balancer can be used. A Load Balancer (LB) distributes clients' requests among underlying Web Servers and makes sure that the load is distributed in an optimal way.

Task

Deploy and configure an Apache Load Balancer for **Tooling Website** solution on a separate Ubuntu EC2 intance. Make sure that users can be served by Web servers through the Load Balancer.

To simplify, let us implement this solution with **2 Web Servers**, the approach will be the same for 3 and more Web Servers.

Prerequisites

Make sure that you have following servers installed and configured within Project-7: (i.e DEVOPS TOOLING WEBSITE SOLUTION)

1. Two RHEL8 Web Servers
2. One MySQL DB Server (based on Ubuntu 20.04)
3. One RHEL8 NFS server

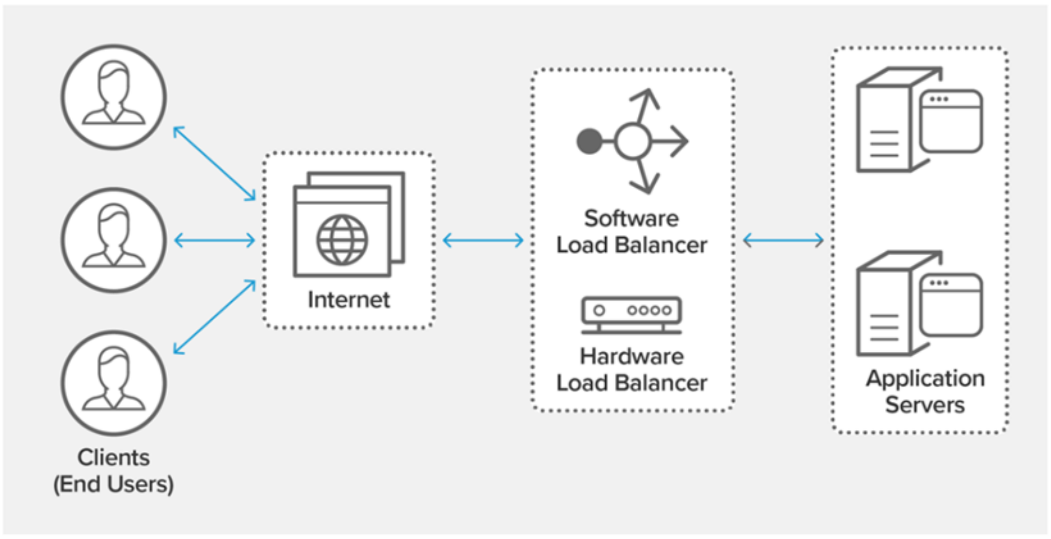
Load balancing refers to efficiently distributing incoming network traffic across a group of backend servers, also known as a *server farm* or *server pool.*

Modern high-traffic websites must serve hundreds of thousands, if not millions, of concurrent requests from users or clients and return the correct text, images, video, or application data, all in a fast and reliable manner. To cost-effectively scale to meet these high volumes, modern computing best practice generally requires adding more servers.

A load balancer acts as the "traffic cop" sitting in front of your servers and routing client requests across all servers capable of fulfilling those requests in a manner that maximizes speed and capacity utilization and ensures that no one server is overworked, which could degrade performance. If a single server goes down, the load balancer redirects traffic to the remaining online servers. When a new server is added to the server group, the load balancer automatically starts to send requests to it.

In this manner, a load balancer performs the following functions:

- Distributes client requests or network load efficiently across multiple servers
- Ensures high availability and reliability by sending requests only to servers that are online
- Provides the flexibility to add or subtract servers as demand dictates

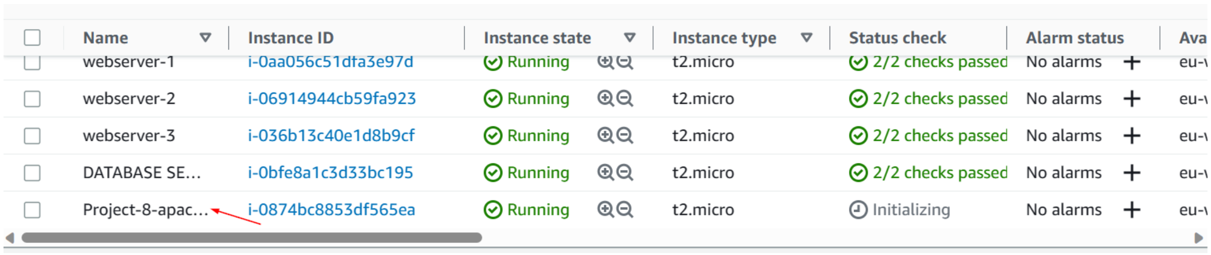


load balancing diagram

CONFIGURE APACHE AS A LOAD BALANCER

Configure Apache As A Load Balancer

1. Create an Ubuntu Server 20.04 EC2 instance and name it `Project-8-apache-lb`, so your EC2 list will look like this:



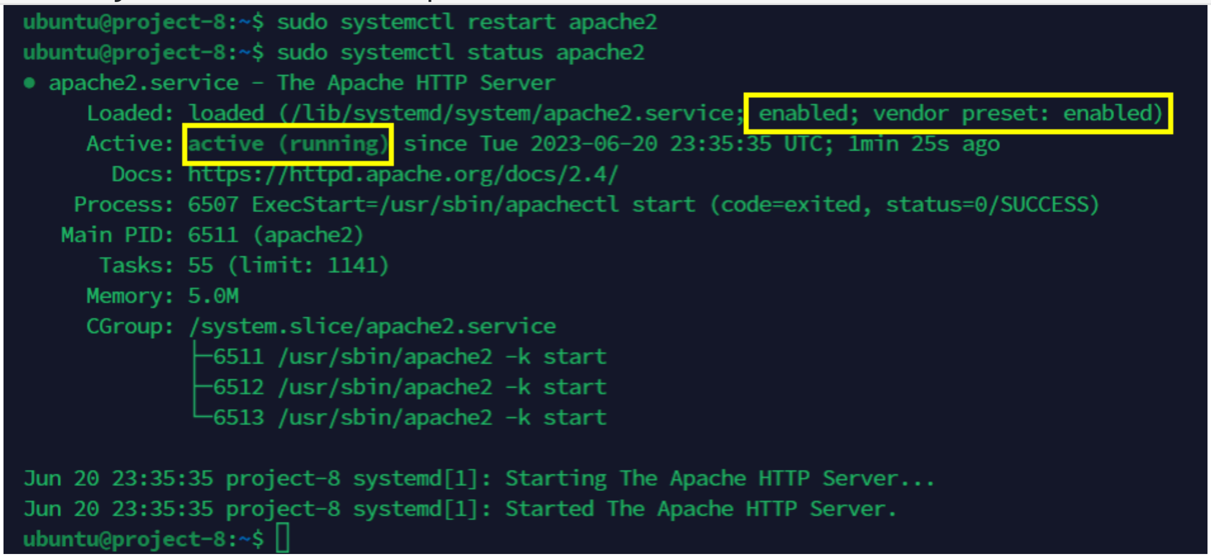
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
☐	webserver-1	i-0aa056c51dfa3e97d	Running	t2.micro	2/2 checks passed	No alarms	eu-\
☐	webserver-2	i-06914944cb59fa923	Running	t2.micro	2/2 checks passed	No alarms	eu-\
☐	webserver-3	i-036b13c40e1d8b9cf	Running	t2.micro	2/2 checks passed	No alarms	eu-\
☐	DATABASE SE...	i-0bfe8a1c3d33bc195	Running	t2.micro	2/2 checks passed	No alarms	eu-\
☐	Project-8-apac...	i-0874bc8853df565ea	Running	t2.micro	Initializing	No alarms	eu-\

2. Open TCP port 80 on `Project-8-apache-lb` by creating an Inbound Rule in Security Group.
3. Install Apache Load Balancer on `Project-8-apache-lb` server and configure it to point traffic coming to LB to both Web Servers:

```
#Install apache2
sudo apt update
sudo apt install apache2 -y
sudo apt-get install libxml2-dev

#Enable following modules:
sudo a2enmod rewrite
sudo a2enmod proxy
sudo a2enmod proxy_balancer
sudo a2enmod proxy_http
sudo a2enmod headers
sudo a2enmod lbmethod_bytraffic

#Restart apache2 service
sudo systemctl restart apache2
```

Make sure apache2 is up and running

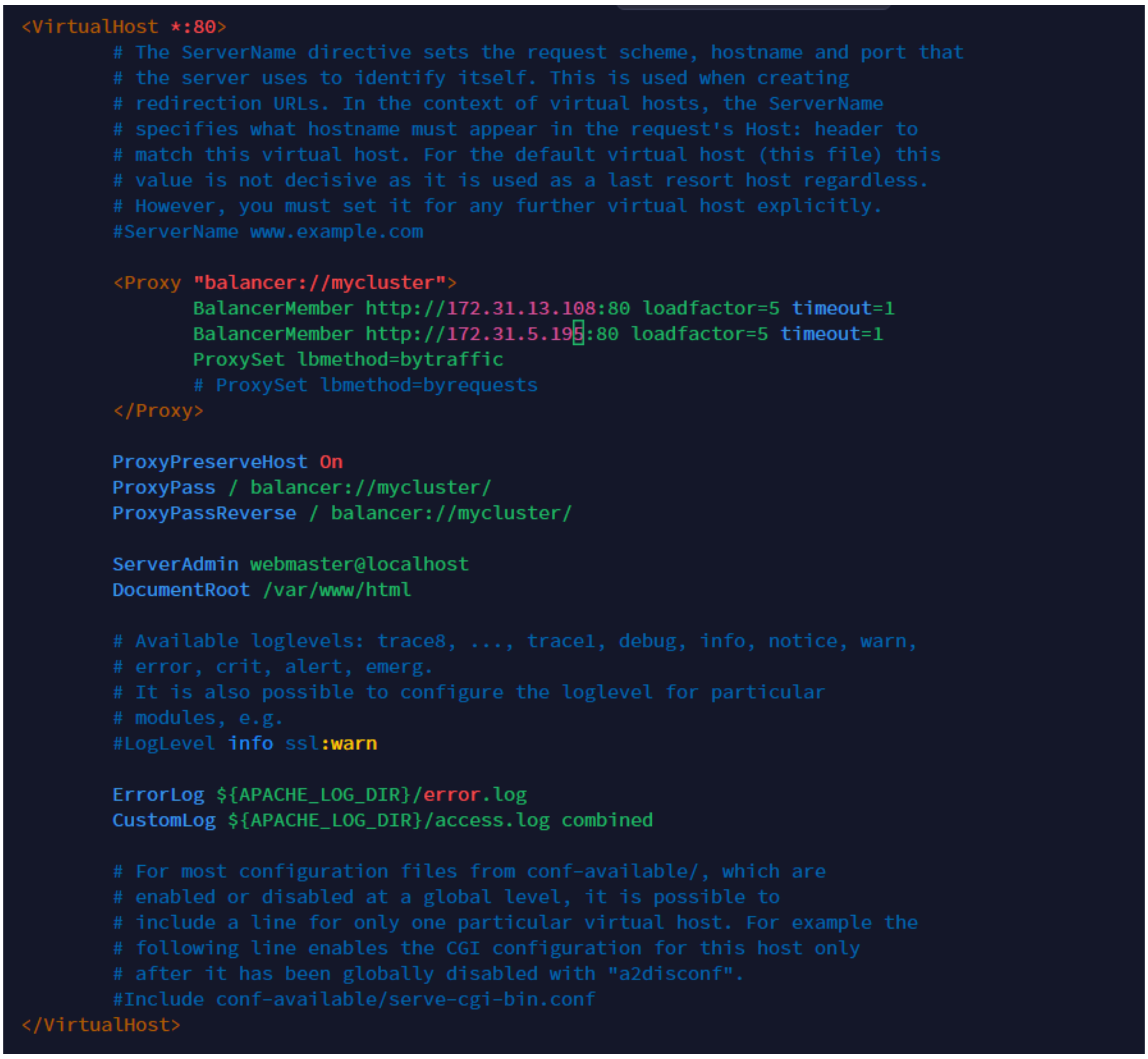
```
sudo systemctl status apache2
```



```
ubuntu@project-8:~$ sudo systemctl restart apache2
ubuntu@project-8:~$ 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 Tue 2023-06-20 23:35:35 UTC; 1min 25s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 6507 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 6511 (apache2)
      Tasks: 55 (limit: 1141)
     Memory: 5.0M
     CGroup: /system.slice/apache2.service
             ├─6511 /usr/sbin/apache2 -k start
             ├─6512 /usr/sbin/apache2 -k start
             └─6513 /usr/sbin/apache2 -k start

Jun 20 23:35:35 project-8 systemd[1]: Starting The Apache HTTP Server...
Jun 20 23:35:35 project-8 systemd[1]: Started The Apache HTTP Server.
ubuntu@project-8:~$ 
```

Configure load balancing

```
sudo vi /etc/apache2/sites-available/000-default.conf

#Add this configuration into this section <VirtualHost *:80>
</VirtualHost>

<Proxy "balancer://mycluster">
               BalancerMember http://<WebServer1-Private-IP-
Address>:80 loadfactor=5 timeout=1
               BalancerMember http://<WebServer2-Private-IP-
Address>:80 loadfactor=5 timeout=1
               ProxySet lbmethod=bytraffic
               # ProxySet lbmethod=byrequests
        </Proxy>

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

#Restart apache server

sudo systemctl restart apache2
```



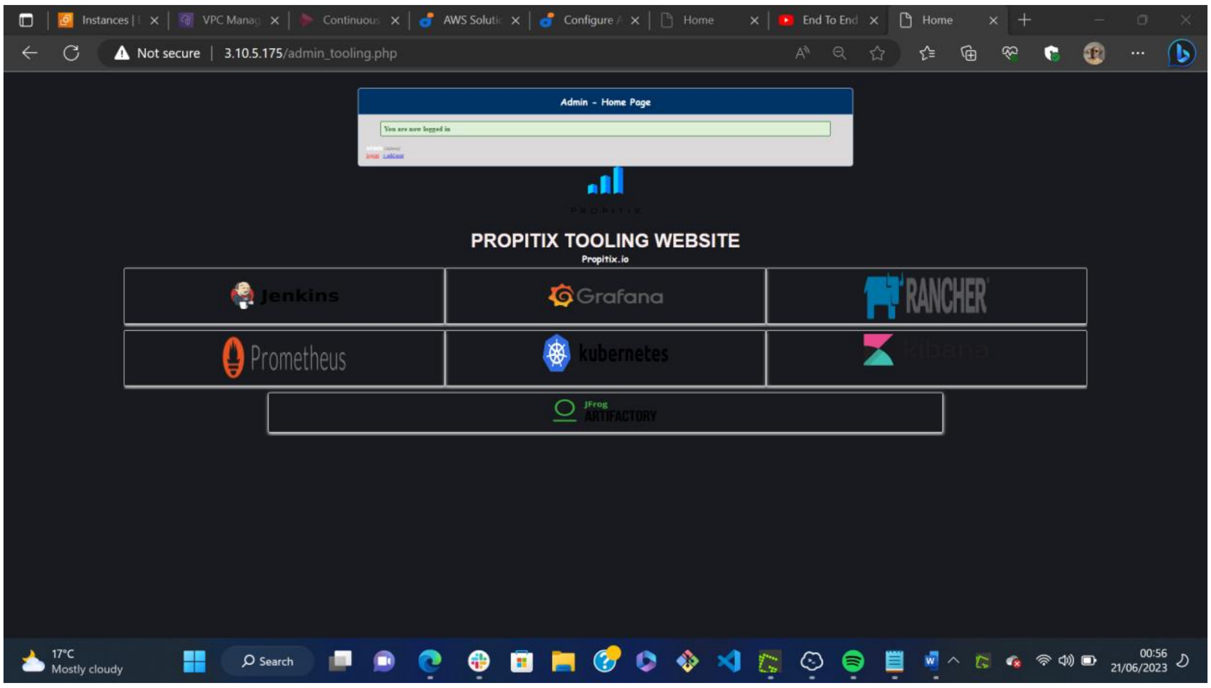
```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        <Proxy "balancer://mycluster">
               BalancerMember http://172.31.13.108:80 loadfactor=5 timeout=1
               BalancerMember http://172.31.5.195:80 loadfactor=5 timeout=1
               ProxySet lbmethod=bytraffic
               # ProxySet lbmethod=byrequests
        </Proxy>

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

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

`bytraffic` balancing method will distribute incoming load between your Web Servers according to current traffic load. We can control in which proportion the traffic must be distributed by `loadfactor` parameter.

You can also study and try other methods, like: `bybusyness`, `byrequests`, `heartbeat`

4. Verify that our configuration works – try to access your LB's public IP address or Public DNS name from your browser:

```
http://<Load-Balancer-Public-IP-Address-or-Public-DNS-Name>/index.php
```



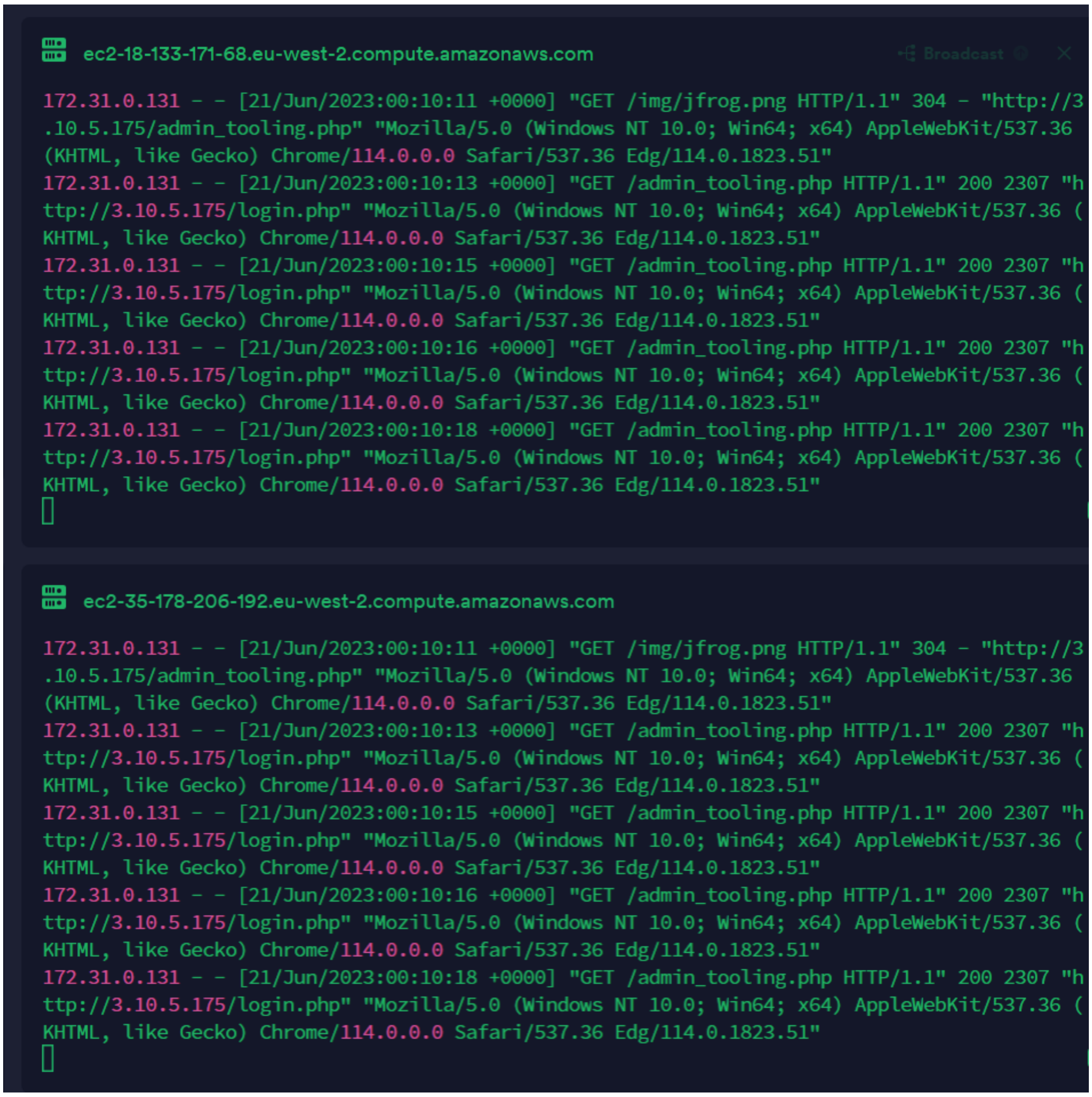
Note: If in the Project-7 you mounted `/var/log/httpd/` from your Web Servers to the NFS server – unmount them and make sure that each Web Server has its own log directory.

Open two ssh/Putty consoles for both Web Servers and run following command:

```
sudo tail -f /var/log/httpd/access_log
```

Try to refresh your browser page `http://<Load-Balancer-Public-IP-Address-or-Public-DNS-Name>/index.php` several times and make sure that both servers receive HTTP GET requests from your LB – new records must appear in each server's log file. The number of requests to each server will be approximately the same since we set `loadfactor` to the same value for both servers – it means that traffic will be disctributed evenly between them.

If you have configured everything correctly – your users will not even notice that their requests are served by more than one server.



```
ec2-18-133-171-68.eu-west-2.compute.amazonaws.com

172.31.0.131 - - [21/Jun/2023:00:10:11 +0000] "GET /img/jfrog.png HTTP/1.1" 304 - "http://3
.10.5.175/admin_tooling.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:13 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:15 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:16 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:18 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"

ec2-35-178-206-192.eu-west-2.compute.amazonaws.com

172.31.0.131 - - [21/Jun/2023:00:10:11 +0000] "GET /img/jfrog.png HTTP/1.1" 304 - "http://3
.10.5.175/admin_tooling.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:13 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:15 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:16 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
172.31.0.131 - - [21/Jun/2023:00:10:18 +0000] "GET /admin_tooling.php HTTP/1.1" 200 2307 "h
ttp://3.10.5.175/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36 Edg/114.0.1823.51"
```

Optional Step – Configure Local DNS Names Resolution

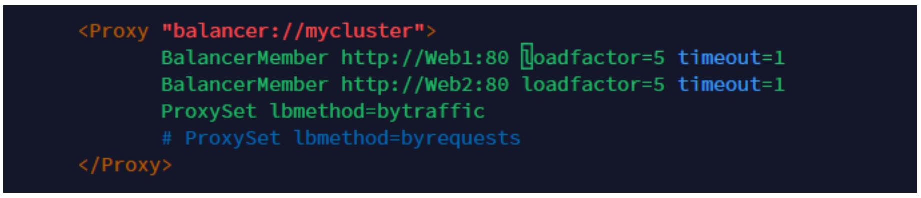
Sometimes it is tedious to remember and switch between IP addresses, especially if you have a lot of servers under your management.
What we can do, is to configure local domain name resolution. The easiest way is to use `/etc/hosts` file, although this approach is not very scalable, but it is very easy to configure and shows the concept well. So let us configure IP address to domain name mapping for our LB.

```
#Open this file on your LB server

sudo vi /etc/hosts
```

```
#Add 2 records into this file with Local IP address and arbitrary
name for both of your Web Servers

<WebServer1-Private-IP-Address> Web1
<WebServer2-Private-IP-Address> Web2
```

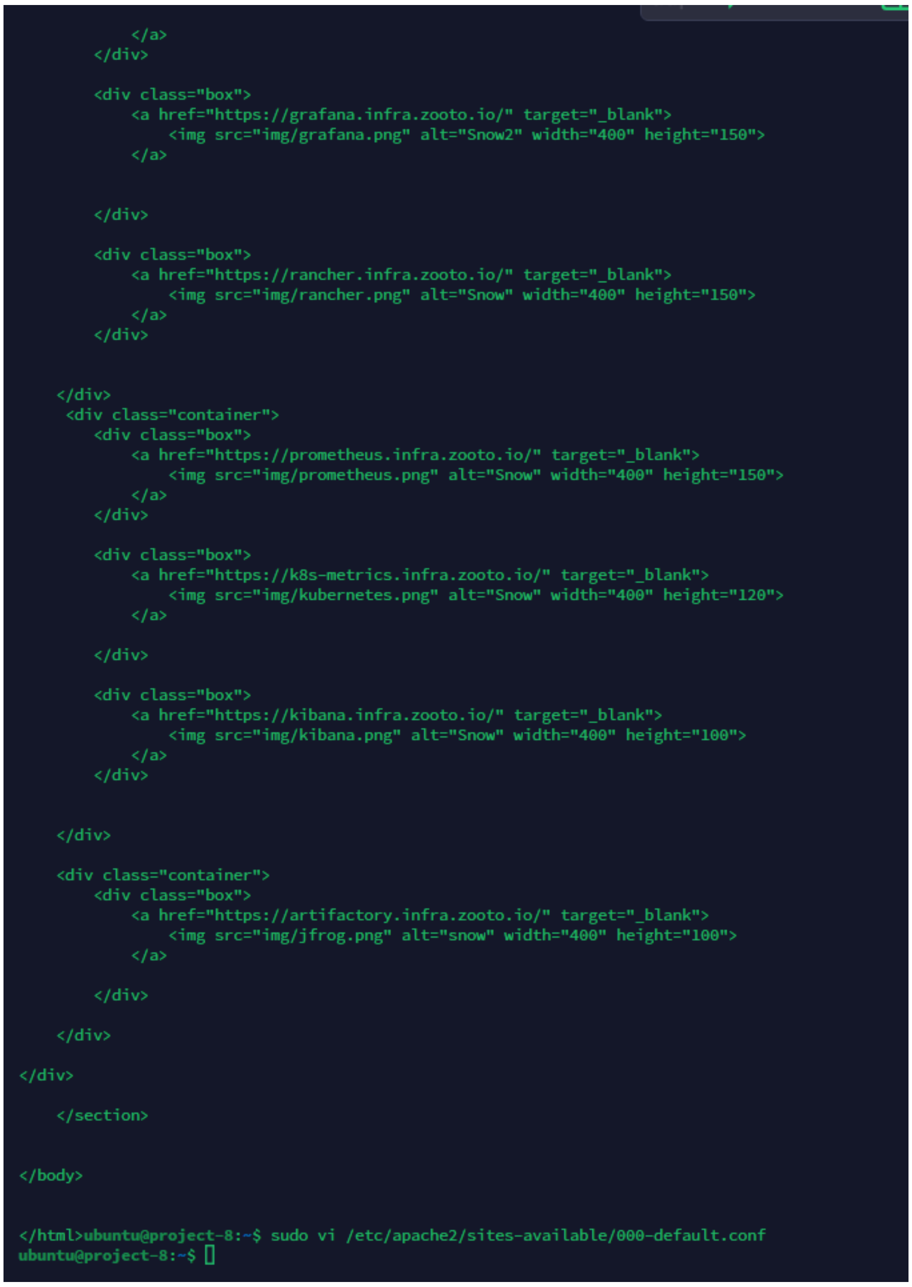
Now you can update your LB config file with those names instead of IP addresses.

```
BalancerMember http://Web1:80 loadfactor=5 timeout=1
BalancerMember http://Web2:80 loadfactor=5 timeout=1
```



```
    <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>
```

You can try to `curl` your Web Servers from LB locally `curl http://Web1` or `curl http://Web2` – it shall work.



```
            </a>
        </div>

        <div class="box">
            <a href="https://grafana.infra.zooto.io/" target="_blank">
                <img src="img/grafana.png" alt="Snow2" width="400" height="150">
            </a>


        </div>

        <div class="box">
            <a href="https://rancher.infra.zooto.io/" target="_blank">
                <img src="img/rancher.png" alt="Snow" width="400" height="150">
            </a>
        </div>


    </div>
     <div class="container">
        <div class="box">
            <a href="https://prometheus.infra.zooto.io/" target="_blank">
                <img src="img/prometheus.png" alt="Snow" width="400" height="150">
            </a>
        </div>

        <div class="box">
            <a href="https://k8s-metrics.infra.zooto.io/" target="_blank">
                <img src="img/kubernetes.png" alt="Snow" width="400" height="120">
            </a>

        </div>

        <div class="box">
            <a href="https://kibana.infra.zooto.io/" target="_blank">
                <img src="img/kibana.png" alt="Snow" width="400" height="100">
            </a>
        </div>


    </div>

    <div class="container">
        <div class="box">
            <a href="https://artifactory.infra.zooto.io/" target="_blank">
                <img src="img/jfrog.png" alt="snow" width="400" height="100">
            </a>

        </div>

    </div>

</div>

    </section>


</body>


</html>ubuntu@project-8:~$ sudo vi /etc/apache2/sites-available/000-default.conf
ubuntu@project-8:~$ 
```

Remember, this is only internal configuration, and it is also local to your LB server, these names will neither be ‘resolvable’ from other servers internally nor from the Internet.

Congratulations!

You have just implemented a Load balancing Web Solution for your DevOps team.