Proyecto Final

Balanceador de Carga con Nginx

CARLOS ANDRES RODRIGUEZ
Facultad de Ingeniería
Universidad Autónoma de Occidente
Cali, Colombia
carlos_a.rodriguez_s@uao.edu.co

JUAN CAMILO AGREDO
Facultad de Ingeniería
Universidad Autónoma de Occidente
Cali, Colombia
juan_camilo.agredo@uao.edu.co

SERGIO BOLAÑOS RAMIREZ
Facultad de Ingeniería
Universidad Autónoma de Occidente
Cali, Colombia
sergio.bolanos@uao.edu.co

***Abstract*—** The following report presents the development of the course project, where evidence of the progress and challenges that are being made in each of the deliveries, where each of the concepts and techniques seen during the classes are implemented, where we start from the structural analysis, determining each of the components to be used, this in order to ensure compliance with the objective and present a device that meets the established specifications.

Keywords- *Structural Analysis, Established Specifications.*

I. Introducción

En el presente documento se realiza una análisis para la construcción de un sistema que permita llevar a cabo el control de carga y eficiencia de conexión de los diferentes clientes a un página web o servidor.

II. Marco teórico

Network Load Balancer:

Un network load balancer actúa como la cuarta capa del modelo de interconexión de sistemas abiertos (OSI). Puede atender millones de solicitudes por segundo. Una vez que el balanceador de carga ha recibido una solicitud de conexión, selecciona un destino en el grupo de destino para la regla predeterminada. Intenta abrir una conexión TCP con el destino seleccionado en el puerto especificado en la configuración de agente de escucha.

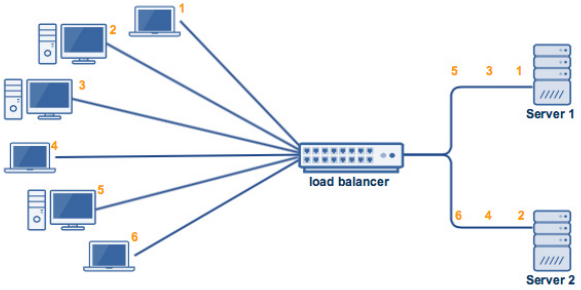


Figura 1. Diagrama de balanceo de carga.

Cuando se habilita una zona de disponibilidad para el balanceador de carga, Elastic Load Balancing crea en ella un nodo de balanceador de carga. De manera predeterminada, cada nodo del balanceador distribuye el tráfico entre los destinos registrados en su zona de disponibilidad solamente. Si habilita el balanceo de carga entre zonas, cada nodo del balanceador distribuye el tráfico equitativamente en todas las zonas de disponibilidad habilitadas.
.Existen diferentes tipos de balanceo de carga entre los cuales tenemos:

- Balanceo de tráfico web: es uno de los casos más habituales, donde se recibe una determinada cantidad de tráfico (por ejemplo tráfico web), y que se reparte entre varios servidores web(basado en apache o Nginx por ejemplo) para atender muchas más visitas de las que podría soportar un único servidor, aunque pudiera aumentarse los recursos al máximo (escalado vertical) de la máquina por ejemplo CPU y RAM, a diferencia de añadir nuevos servidores para atender peticiones (escalado horizontal).
- Balanceo de bases de datos: También existen diferentes soluciones que permiten balancear peticiones entre varios servidores diferentes. En MySQL, hay varias opciones dependiendo de si se necesita un entorno multimaster o con varios de solo lectura, como por ejemplo Galera Cluster o MySQL Cluster son de las opciones más recomendables.
- Balanceador de comunicaciones: Cuando se tienen varias líneas de comunicaciones que permiten tener conectividad hacia un determinado destino, que puede ser internet o hacia una plataforma CLOUD. Al tener varios canales de comunicaciones, se necesita un dispositivo que se encargue de repartir el tráfico entre esos enlaces, de forma activa/activa repartiendo la carga en base a algún patrón predefinido (% del tráfico, por IPs destino, por tipo de tráfico, …) o también se puede utilizar como solución activa/pasiva para proporcionar

alta disponibilidad en caso de caída del enlace principal.

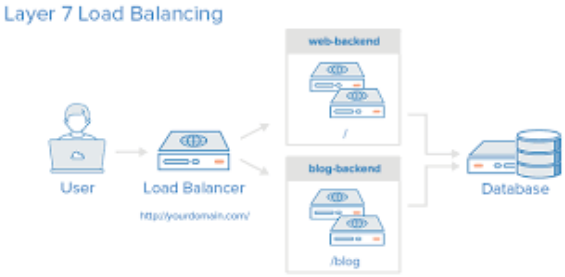


Figura 2. Distribución del balanceo de carga.

Existen diferentes métodos de balanceo que definen cómo se reparten las sesiones entre cada uno de los frontales. Donde se pueden destacar las siguientes como las habituales:

- Robin Round: las peticiones se reparten entre los frontales disponibles de una en una desde el primero hasta el último, de forma circular. De esta forma todos los frontales recibirán exactamente el mismo número de peticiones. Si se tienen diferentes capacidades de CPU o recursos en los diferentes servidores este método no sería muy adecuado.

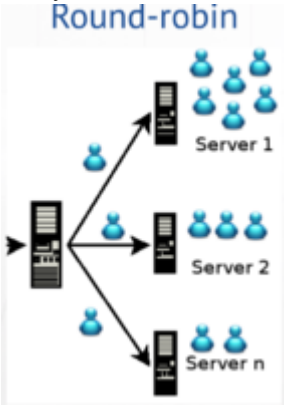


Figura 3. Diagrama de Round -Robin.

- Weighted Robin Round: cada servidor tiene un determinado peso en el reparto que indica qué porcentaje de las peticiones que se entregarán a cada servidor. Con esto se intenta identificar previamente que servidores podrán procesar más carga bien por que dispongan de más recursos o están mejor optimizados para procesar las peticiones.

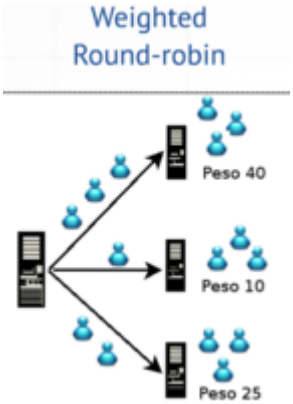


Figura 4. Diagrama de Weighted Round-robin.

- Least Connections: las peticiones se balancean al servidor con el menor número de conexiones establecidas en ese preciso momento. De esta forma el servidor menos cargado atenderá las nuevas peticiones pero solamente en base al número de peticiones, y no se garantiza que siempre se conecte al menos cargado ya que en ocasiones se exigen mayor capacidad de procesamiento, memorias u otros recursos.

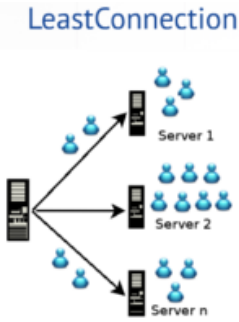


Figura 5. Diagrama de Least Connection.

- IP-Hash: la primera conexión que llega de una determinada dirección IP se balancea al frontal que corresponda (habitualmente por Robin round), y se apunta en una tabla esa asignación. Desde ese momento las nuevas conexiones de esa misma ip se balancean en ese mismo frontal mantiene la resistencia por ip de origen.

III. CONFIGURACIÓN

Para configurar nuestro balanceador de carga lo primero que se debe tener en cuenta es que es necesario tener instalados los servicios de HTTP de la siguiente forma:

yum install httpd -y

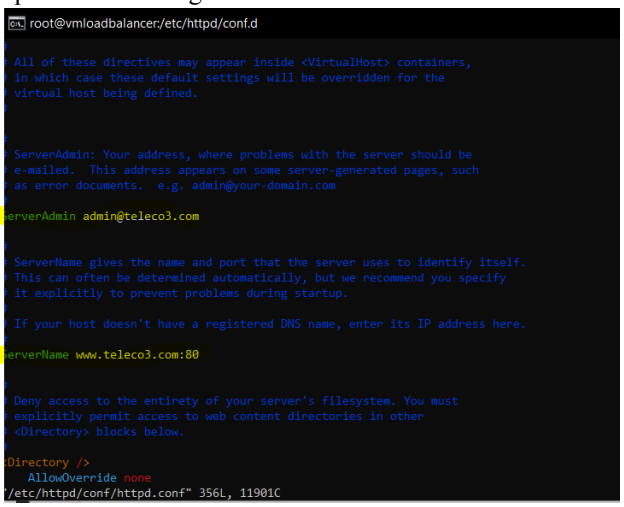
se debe iniciar el servicio para verificar que los paquetes se hayan instalado correctamente:

systemctl start httpd

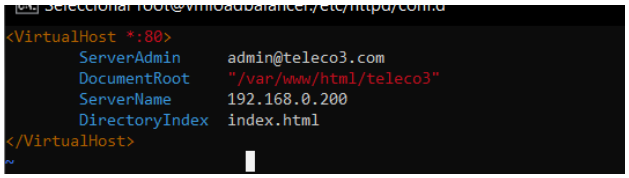
Pasamos a configurar el httpd.conf

vim /etc/httpd/conf/httpd.conf

aquí se debe configurar nuestra red en nuestro caso



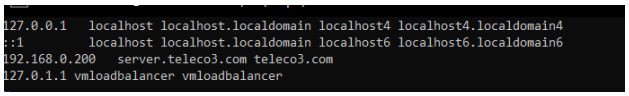
vim /etc/httpd/conf.d/teleco3-host.conf



se configura la zona virtual
Se procede a reiniciar el servicio
systemctl restart httpd

se configura el host en el loopback

vim /etc/hosts



Se crea la plantilla en la ruta del index con el nombre del servicio, una pequeña página html para visualizar en el navegador.

Ahora procedemos a configurar el servicio del NGINX

--BALANCEADOR:

1. Levantar las máquinas

vagrant up

2. conectarse al balanceador

vagrant ssh balancer

3. Instalar nginx

sudo yum install -y nginx

4. Ir a la ruta y crear dos carpetas

cd /etc/nginx/
sudo mkdir sites-available
sudo mkdir sites-enabled

5. Crear el siguiente archivo

sudo vi /etc/nginx/sites-available/load-balancer
#Añadir el siguiente contenido
upstream backend {
server webapache_1;
server webapache_2;
server webapache_n
}

server {
listen 80;
listen [::]:80;
server_name balancer_server;
proxy_redirect off;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $http_host;
location / {
proxy_pass http://backend;
}
}

6. abrir el archivo de configuración de nginx

sudo vi /etc/nginx/nginx.conf

#Añadir el siguiente contenido
include /etc/nginx/sites-enabled/*;

7. crearemos un enlace simbólico del archivo load-balancer creado anteriormente para que el servidor Nginx lo logre leer

sudo ln -s /etc/nginx/sites-available/load-balancer /etc/nginx/sites-enabled/

8. Reiniciar el servicio nginx.

systemctl reload nginx

IV. Otras posibles soluciones

Para configurar un servidor usando HAProxy se debe realizar la siguiente serie de pasos.

yum install haproxy

Una vez instalado, conviene realizar una copia de seguridad del fichero original del producto, por si metemos la pata cuando configuremos nuestros servicios. El fichero es: */etc/haproxy/haproxy.cfg.*

Configuración de HAProxy

Supongamos que tenemos dos servidores WEB escuchando por las IPs:

1. 192.168.1.2:443
2. 192.168.1.3:443

Y queremos balancear la carga entre ambos servidores. En el fichero */etc/haproxy/haproxy.cfg*, configuraremos las IPs y puertos por donde escuchan los servicios, en concreto, la sección *listen*:

```
listen ServidorWeb
   bind *:80
   mode http
   stats enable
   stats auth  cda:cda
   balance roundrobin
   server frontend-01 192.168.1.2:80 check weight 40
   server frontend-02 192.168.1.3:80 check weight 60
```

Como podemos ver en el ejemplo anterior, balanceamos el servicio por roundrobin y distribución de pesos en relación 40-60%. Roundrobin significa que una petición va a un servidor y la siguiente a otro (teniendo en cuenta la distribución de pesos)

Arranque de HAProxy

En el caso de Linux CentOS, arrancaremos y habilitaremos el servicio con el arranque del sistema con los siguientes comandos:

```
systemctl start haproxy
systemctl enable haproxy
```

Una vez que tenemos arrancado el servicio, podemos acceder a las **estadísticas** de balanceo con una URL interna de HAProxy, por ejemplo:

http://192.168.1.4:8080/stats

V. Conclusiones

Para la creación de las máquinas virtuales del cluster de servidores resulta muy útil las herramientas de Vagrant para importar o aprovisionar las máquinas; en nuestro caso se utiliza un box de VagrantCloud creado por Juan Camilo, (Revisar Vagrantfile), esto facilita la creación de muchas máquinas idénticas con el servicio ya instalado en este caso http con Apache, las cuales fueron utilizadas en la práctica.

En nuestro caso elegimos Nginx como FrontEnd para recibir las peticiones, ya que este es un servicio web que nos permite fácilmente configurarlo como balanceador de carga y de una manera sencilla, además es un servidor gratuito, código abierto y ofrece un alto rendimiento.

Es muy importante identificar y planear como se desean distribuir las conexiones y la comunicación entre los distintos clientes y servidores para utilizar un algoritmo adecuado que nos permita tener la robustez necesaria para mantener el servicio sin que este se pierda o caiga, en nuestro caso particular recurrimos a la lógica de Round-robin ya que las consultas a realizar no iban a deterioran rápidamente la red y no soportaria tanto tráfico de datos.

VI. Referencias

[1] Balanceo de carga.(s.f). RedesTeleco.Recuperado 16 de mayo de 2022, de https://redesteleco.com/balanceo_de_carga/

[2] ¿Qué es el balanceo de carga? (2013,23 enero). IDG Communications S.A.U. https://www.computerworld.es/tendencias/que-es-el-balanceo-de-carga

[3] Que es un balanceador de carga de red? - Elastic Load Balancing. (s.f). Amazon Web Service. Recuperado 13 de mayo de 2022, de https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/network/introduction.html