

Balanceador de carga con NGINX

David Aristizabal, Manuel Tarazona, Christiam Vera, Camilo González
 Universidad Autónoma de Occidente
 2021

Resumen – En el presente documento se mostrarán algunas tecnologías que permiten la implementación de un balanceo de carga para un servidor web, mostrando algunas de sus características y mostrando una implementación sencilla de este proceso usando Nginx, con las conclusiones debido al proceso de este trabajo.

Palabras clave – Balanceo de carga, Nginx, HAProxy, servidor web.

I. INTRODUCCIÓN

Actualmente al momento de realizar trabajos con servidores se requieren servicios de máquinas virtuales y uno de los elementos más importantes son los balanceadores de carga que, como su nombre lo indica distribuyen la carga de manera automática entre varios destinos ya sean estos servidores o contenedores. El balanceador de carga o load balancer fundamentalmente es un dispositivo software o hardware que se encarga de revisar el estado de los destinos y redirigirlo a los que estén en buen estado o cumplan algunos requisitos [1].

Estos se pueden dividir en balanceadores de carga estáticos y dinámicos, diferenciándose en el hecho de que el estático se realiza antes de la ejecución de cualquier proceso y el dinámico ejecuta la acción durante la ejecución de los procesos, indicando que la división computación depende de la cantidad de tareas o peticiones se estén realizando[2].

También se tiene el balanceador de carga elástico el cual se adapta automáticamente al tráfico realizando un auto escalado para cumplir con las peticiones solicitadas. El balanceador de carga es el único punto de contacto con los clientes, ya que este se encarga de redirigir el tráfico y peticiones a las distintas direcciones o backends.

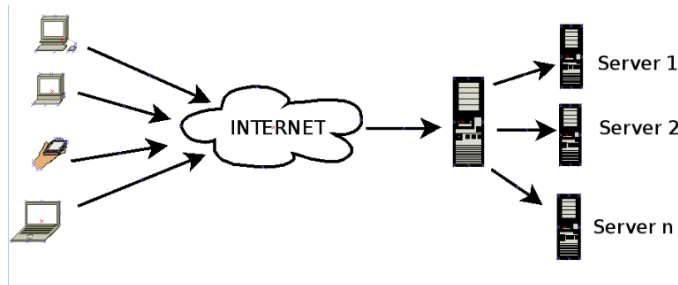


Fig.1 Balanceador de carga [3]

Como se puede apreciar en Fig. 1 el balanceador de carga es el que está recibiendo las peticiones y redirigiéndolo a los demás servidores también llamados backends.

II. TECNOLOGÍAS DE BALANCEADOR DE CARGA

Como ya me mencionó el balanceador de carga puede ser un dispositivo hardware o software y este documento se enfocará explorando algunas alternativas o tecnologías usadas tales como las siguientes:

A. Tipo hardware

Este consiste en un servidor especializado junto a un sistema operativo en específico, además, un software para hacer el proceso de balanceador de carga. Este servidor reúne los servidores web mediante soluciones Plug and Play, lo que significa que una vez conectado, funcionan con el mínimo de ajustes previos.

B. Tipo switch

Este tipo de balanceadores de carga necesita de un switch Layer 2 o Layer 3 para la implementación del proceso de balanceo. No se hace necesaria la conexión de ningún dispositivo intermediario entre el switch y el servidor web. Un switch es un dispositivo que reenvía el tráfico (dado en tramos) en función de la información contenida en esta capa (Al estar en la capa tres esta ofrece tecnologías de enrutamiento, conmutación y reenvío)

C. Basado en software

Por último, el balanceador de carga basado en software, se realiza la instalación de este y se administra y configura el software. Esto puede ser comercial o de código abierto/gratuito y a continuación se mostrarán algunos casos:

D. HAProxy

HAProxy es un balanceador de carga que está enfocado netamente en esta función, utilizando TCP o http, es una solución de código abierto desarrollado en C [4]. Dentro de sus funciones se pueden encontrar el balanceo de carga o el enrutamiento de algunos tráfico hacia otros servidores y

también puede traducir tráfico desde https hacia http para una mejor compresión de la aplicación [5].

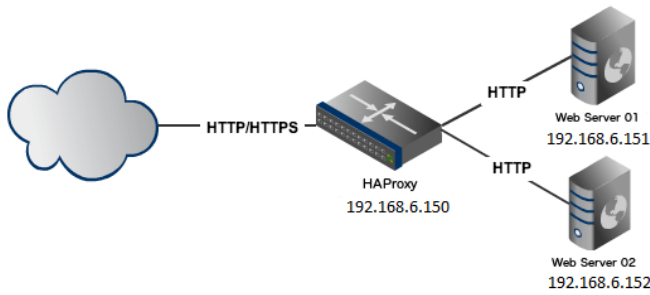


Fig. 2 Balanceador de carga con HAProxy

Como se puede apreciar en la Fig. 2 se tiene una implementación de un balanceador de carga con haproxy manteniendo la configuración mostrada en la Fig. 1. Cabe resaltar se puede configurar el haproxy para que realice la distribución o balanceo dependiendo de un algoritmo como Round Robin, Least connections, IP Hash, Weighted, entre otros.

E. Pen

Pen es un balanceador de carga muy simple que permite tener varios servidores en línea entregando un servicio y que públicamente aparezcan como uno. Una de sus funcionalidades es detectar los servidores que no están disponibles y distribuir la carga entre los que están, con una alta disponibilidad y rendimiento con alta escalabilidad.

F. Nginx

Es un servidor web de código abierto que también puede ser usado como proxy inverso y realiza operaciones de balanceo de carga [6]. Fue creado en el año 2004 y se basa en la optimización de rendimiento bajo escala. Nginx está diseñado para uso de baja memoria y alta concurrencia.

Nginx es el segundo servidor web más popular después de apache aunque es el más popular entre los sitios web con mucho tráfico[6]. Se puede controlar con un proceso maestro múltiples procesos esclavos, permite implementar balanceadores de carga de manera sencilla usando módulos `http_upstream` y `http_proxy`[2].

Dentro de las fortalezas de Nginx está el que es un software rentable, es de código abierto, muy popular entre desarrolladores adecuado para contenedores y en usos de proxy inverso y servidor web de alto rendimiento.

G. Seesaw

En un balanceador de carga utilizado por Google, confiable basado en Linux para distribuir carga en una red. Esta desarrollado en el lenguaje Go y trabaja en distribuciones de

Debian y Ubuntu. Este balanceador de carga funciona sólo en la capa cuatro del modelo OSI de modo que no es posible para implementar en la capa siete de aplicación [7].

H. Traefik

Es un proxy inverso y balanceador de carga integrado con Docker y kubernetes además, de tecnologías de Cluster como Mesos o Amazon ECS [8].

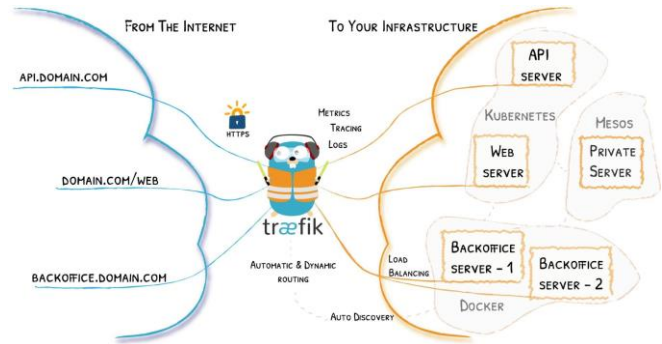


Fig. 3 Configuración de traefik [8]

Traefik es un balanceador de carga adaptado a la computación en la nube mediante microservicios. Este se integra con los principales componentes de infraestructura configurándose a sí mismo automáticamente y de manera dinámica. Traefik es muy fácil de operar siendo capaz de manejar sistemas complejos y grandes en diversos entornos y diferentes capas de la pila de red como HTTP, TCP o UDP.

III. IMPLEMENTACIÓN

Para este proyecto se requiere implementar un balanceador de carga y para ello se utilizó NGINX. Para ello se crean tres máquinas virtuales en VirtualBox, de modo que en máquina número uno se instaló la herramienta NGINX y las otras dos se utilizaron en forma de backend de modo que permita recibir todas las solicitudes que se le realizan al servidor, para ello se instala NGINX en las tres máquinas permitiendo inicializar un servicio de http y solo deja la necesidad para su correcto funcionamiento, cambiar en cada backend el contenido del `index.html`.

En el servidor principal el cual contiene el balanceador de carga, se debe crear un archivo con nombre `load-balancing`, donde deben añadir las IP de los servidores que van a funcionar como backend. Se implementó el backend 1 y 2 en la parte del upstream donde se añadió el método `ip_hash`, el cual basado en un algoritmo que tiene este método, asigna un enrutamiento para cada cliente.

El servidor se colocó a escuchar por el puerto 80, habilitándose para llamar desde dos dominios (`server.pjfinal.com`, www.pjfinal.com) además de ser posible esta acción con la IP.

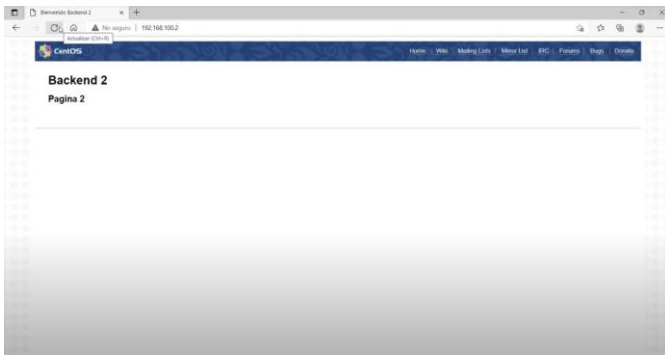


Fig. 4 Página de backend 2

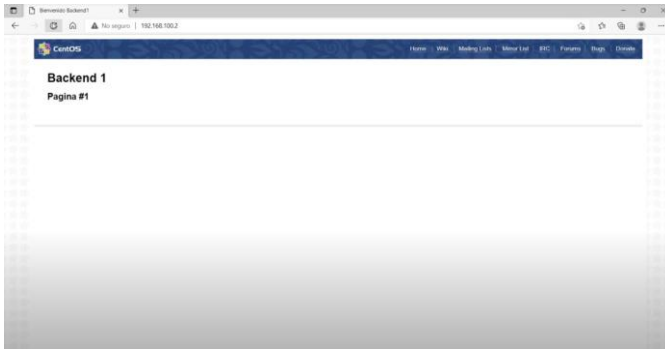


Fig. 5 Página de backend 1

Para la verificación del correcto funcionamiento, desde el navegador se realiza una petición al balanceador de carga y este realiza la redirección dependiendo del método de balanceo de carga configurado como se puede observar en Fig 4 y Fig. 5.

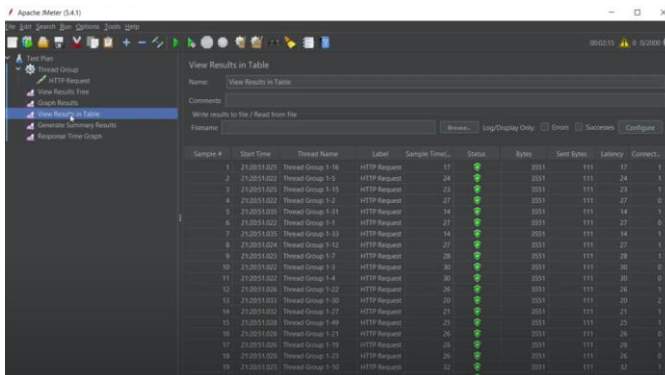


Fig. 6 Prueba de carga con JMeter.

Se realizaron varias pruebas donde se asignaron dos mil usuarios para que hicieran una petición de HTTP como se muestra en la Fig. 6 y en se logra apreciar en la Fig. 7 el tiempo de carga para dichos casos.

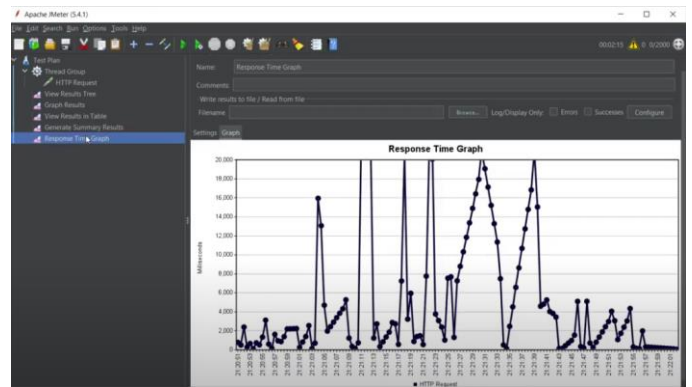


Fig. 7 Tiempo de respuesta de las peticiones en JMeter

En la gráfica anterior podemos ver el tiempo de respuesta que genera el balanceador de carga y además podemos evidenciar que en algunos puntos son muy altos ya que posiblemente las IP de los clientes tienden a asignar un servidor en específico esto debido a método utilizado para esta prueba.

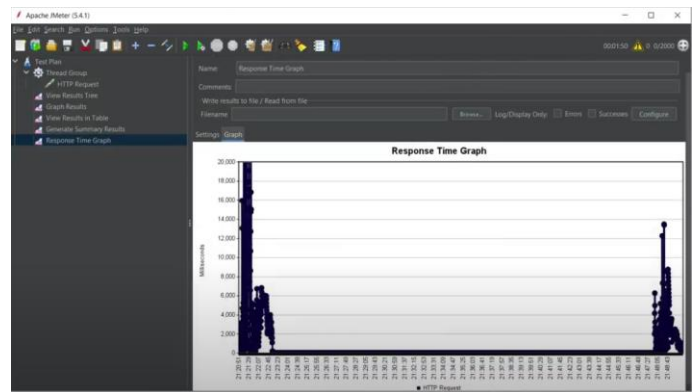


Fig. 8 Tiempo de respuesta del método Round robin

En la Fig. 8 se muestra el comportamiento de las peticiones realizadas con JMeter al balanceador de carga usando el método Round robin, el cual viene por defecto en la configuración de balanceo de carga con NGINX.

CONCLUSIONES

- Para la elaboración de este trabajo se indagaron de manera proporcional las herramientas que permiten configurar un balanceo de carga de manera satisfactoria, aunque se eligió NGINX por su facilidad de implementación, además de ser de código abierto y consumir menos recursos.
- A pesar de las herramientas mostradas durante este proyecto, es importante mencionar que existen en el mercado herramientas que permiten realizar el balanceo de carga junto a muchas otras operaciones de manera automatizada, estas son plataformas o servicios como AWS, Azure, entre otros.
- Implementar un balanceador de carga por medio de software tiene la ventaja de ser muy económico, relativamente sencillo y fácil de usar.

REFERENCIAS

- [1] “¿Qué es un Application Load Balancer? - Elastic Load Balancing”.
https://docs.aws.amazon.com/es_es/elasticloadbalancing/latest/application/introduction.html (consultado nov. 06, 2021).
- [2] R. X. P. Aragonés, “SOD ROBERTO PORTES”,
 Consultado: nov. 07, 2021. [En línea]. Disponible en:
https://www.academia.edu/35080992/SOD_ROBERTO_PORTES
- [3] “Equilibrador de carga”, *Wikipedia, la enciclopedia libre*.
 nov. 02, 2020. Consultado: nov. 06, 2021. [En línea].
 Disponible en:
https://es.wikipedia.org/w/index.php?title=Equilibrador_de_carga&oldid=130566554
- [4] codigofacilito, *Balanceador de carga con HAproxy*,
 (2019). Consultado: nov. 06, 2021. [En línea Video].
 Disponible en:
<https://www.youtube.com/watch?v=7ljiOD6MbLA>
- [5] Achetronic, *HAproxy con SSL en 5 minutos con Docker*,
 (sep. 27, 2020). Consultado: nov. 06, 2021. [En línea Video].
 Disponible en:
<https://www.youtube.com/watch?v=twCmZfSSWwc>
- [6] “¿Qué Es Nginx y Cómo Funciona?”, *Kinsta*.
<https://kinsta.com/es/base-de-conocimiento/que-es-nginx/>
 (consultado nov. 07, 2021).
- [7] “10 Open Source Load Balancer for HA and Improved Performance”, *Geekflare*, oct. 29, 2017.
<https://geekflare.com/open-source-load-balancer/>
 (consultado nov. 07, 2021).
- [8] “Traefik: Proxy inverso y balanceador de carga para Docker”, *Cloud Center Andalucía*, feb. 02, 2021.
<https://www.cloudcenterandalucia.es/blog/traefik-proxy-inverso-y-balanceador-de-carga-parte1/> (consultado nov. 07, 2021).