



Balanceo de Carga
Y otros súper poderes

¿Quién?

Jorge Alejandro Cajas Mota
jac.mota@gmail.com



guate-jug



Comunidad Desarrolladores en Tecnologías
< Google > en Guatemala

¿NGINX?

Un servidor http con super poderes

- Fue creado en 2004
- Es multiplataforma
- Es el segundo servidor web más usado
- No ejecuta scripts* , es utilizado como servidor frontal para balanceo de carga, contenido estático o proxy inverso.
- Puede crear servidores virtuales

¿Por qué usar NGINX?

- Servidor basado en eventos, soporte mas clientes activos
- El proxy inverso permite tener servidores diferentes en un mismo dominio
- El balanceo de carga nos permite distribuir el trabajo y tener alta disponibilidad
- Permite configuraciones de seguridad, SSL, TLS
- Podemos implementar una caché estática y mejorar los tiempos de respuesta

Herramientas Disponibles

- Contenido estático
- Proxy Inverso
- Caché estática
- Balanceo de Carga
- Instalación de certificados SSL/TLS
- Compresión de contenido
- Streaming de contenido (FLV, MP4)
- Muchos módulos más

Proxy Inverso

- Es un servidor que recibe las solicitudes de la Internet y hacia delante (proxies) a un pequeño conjunto de servidores, que normalmente se encuentra en una red interna y no directamente accesible desde el exterior.

Proxy Inverso

```
http{

    upstream backend {
        server 127.0.0.1:8080
    }

    upstream blog{
        server 127.0.0.1:4200
    }

    server{
        listen      80 default_server;
        listen      [::]:80 default_server;
        server_name  example.com www.example.com;
        root        /home/server/static;

        location /api {
            proxy_pass http://backend;
        }

        location /blog {
            proxy_pass http://blog;
        }

    }
}
```

Balanceo de Carga

- Con el Balanceo de carga, podemos distribuir las peticiones de nuestro sitio web hacia varios servidores, esto permite que el escalado sea horizontal y podemos agregar / quitar servidores como la carga del sitio lo requiera.
- Podemos usar técnicas para el balanceo como roundrobin, last connection, ip_hash o por peso

Balanceo de Carga

```
http{  
  
    upstream backend {  
        least_conn;  
        server 127.0.0.1:8080  
        server 10.10.1.1:8080  
        server 192.168.1.10:8080  
    }  
  
    upstream blog{  
        server 127.0.0.1:4200  
    }  
  
    server{  
        listen      80 default_server;  
        listen      [::]:80 default_server;  
        server_name  example.com www.example.com;  
        root         /home/server/static;  
  
        location /api {  
            proxy_pass http://backend;  
        }  
  
        location /blog {  
            proxy_pass http://blog;  
        }  
    }  
}
```

Caché estático

- Con el caché estático evitamos que el servidor tenga que leer los archivos desde el disco, en su lugar los mantiene en memoria para responder aun más rápido.

Caché estático

```
http{

    proxy_cache_path /data/nginx/cache keys_zone = myCache : 10m;

    upstream backend {
        least_conn;
        server 127.0.0.1:8080
        server 10.10.1.1:8080
        server 192.168.1.10:8080
    }

    upstream blog{
        server 127.0.0.1:4200
    }

    server{
        listen      80 default_server;
        listen      [::]:80 default_server;
        server_name  example.com www.example.com;
        root         /home/server/static;

        location /api {
            proxy_pass http://backend;
        }

        location /blog {
            proxy_pass http://blog;
        }

    }
}
```

Certificados de Seguridad

- Al utilizar NGINX como servidor frontal, podemos usar un solo certificado de seguridad que afecte a todos los servidores que se encuentran atrás de él, así mantenemos un solo certificado y nos ahorramos tiempo al momento de tener que actualizarlo.

Certificados de Seguridad

```
server {  
    listen      443 ssl;  
    listen      [::]:443 ssl;  
    server_name example.com www.example.com;  
    root        /home/server/static;  
  
    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;  
  
    ...  
  
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
    ssl_prefer_server_ciphers on;  
    ssl_dhparam /etc/ssl/certs/dhparam.pem;  
    ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH"  
    ssl_session_timeout 1d;  
    ssl_session_cache shared:SSL:50m;  
    ssl_stapling on;  
    ssl_stapling_verify on;  
    add_header Strict-Transport-Security max-age=15768000;  
}
```

Ventajas

- Al ser un servidor HTTP que no ejecuta scripts, puede servir contenido más rápido y de manera más eficiente.
- Al colocarlo como servidor frontal podemos no solo manejar varios servidores de diferente tipo atrás de el, también podemos manejar un solo certificado de seguridad.
- Al integrar el balanceo de carga podemos garantizar una alta disponibilidad de nuestro servicio.

Desventajas

- Al ser un servidor frontal necesita tener suficiente ancho de banda para soportar todas las peticiones, aunque estas sean procesadas por un servidor de backend.
- No se encuentra disponible (aun) en la mayoría de hostings económicos, por lo que debemos de recurrir a VPS o similares.
- La configuración, a diferencia de Apache es única, por lo que no existen archivos de configuración a nivel de carpetas como los .htaccess.