Balanceo de carga en el Proxy

AWS

Yeray Gutiérrez Mullor

Contenido

Balanceo de carga:

Cuando finalizamos la configuración del servidor web, decidimos implementar un nuevo servidor web (Clonando el original); Después de clonarlo, teníamos que poner la web con alta disponibilidad, para lo que implementamos balanceo de carga en el Proxy de la instancia NAT

Explicación del código:

proxy.conf

sites-available > proxy.conf

```
# Definición del grupo de servidores con detección de fallos
upstream servidores_web {
    # Si un servidor no responde en 2 segundos, Nginx lo marca como caído
    server 10.0.128.56:80 max_fails=1 fail_timeout=5s;
    server 10.0.128.20:80 max_fails=1 fail_timeout=5s;
}

server {
    listen 80;
    server_name _;

    # Carpeta raíz y archivo índice (opcional si solo haces proxy)
    root /var/www/html;
    index index.php index.html;

    location / {
        # Enviamos el tráfico al grupo definido arriba
        proxy_pass http://servidores_web;

        # Cabeceras corregidas para evitar el Error 400
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # CONFIGURACIÓN DE SALTO AUTOMÁTICO (Failover)
        # Si el servidor da error o está apagado, salta al siguiente de inmediato
        proxy_next_upstream error timeout http_500 http_502 http_503 http_504;

        # Tiempos de espera cortos para que el usuario no note el servidor caído
        proxy_connect_timeout 2s;
        proxy_read_timeout 10s;
        proxy_send_timeout 10s;
    }
}
```

1. Bloque upstream:

```
upstream servidores_web {
    # Si un servidor no responde en 2 segundos, Nginx l
    server 10.0.128.56:80 max_fails=1 fail_timeout=5s;
    server 10.0.128.20:80 max_fails=1 fail_timeout=5s;
```

Este bloque define el grupo de servidores a los que accede el proxy.

upstream servidores_web { ... }: Crea un grupo llamado "servidores_web".

server 10.0.128.56:80 y 10.0.128.20:80: Son las direcciones IP privadas de los servidores finales.

max_fails=1 fail_timeout=5s: Si un servidor falla 1 vez, Nginx lo considerará "caído" durante 5 segundos antes de intentar enviarle tráfico otra vez.

2. Bloque server:

```
server {
    listen 80;
    server_name _;

    # Carpeta raíz y archivo índice (opcional si solo haces proxy)
    root /var/www/html;
    index index.php index.html;

    location / {
        # Enviamos el tráfico al grupo definido arriba
        proxy_pass http://servidores_web;

        # Cabeceras corregidas para evitar el Error 400
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # CONFIGURACIÓN DE SALTO AUTOMÁTICO (Failover)
        # Si el servidor da error o está apagado, salta al siguiente de inmediato
        proxy_next_upstream error timeout http_500 http_502 http_503 http_504;

        # Tiempos de espera cortos para que el usuario no note el servidor caído
        proxy_connect_timeout 2s;
        proxy_read_timeout 10s;
        proxy_send_timeout 10s;
    }
}
```

```
server {
    listen 80;
    server_name _;

    # Carpeta raíz y archivo índice
    root /var/www/html;
    index index.php index.html;
```

Aquí se define cómo escucha Nginx las peticiones externas.

listen 80;: Nginx escucha en el puerto estándar HTTP.

server_name _;: significa que responderá a cualquier dominio o IP que apunte a este servidor.

root e index: Definen la carpeta base y los archivos por defecto.

3. Bloque location:

```
location / {
    # Enviamos el tráfico al grupo definido arriba
    proxy_pass http://servidores_web;

    # Cabeceras corregidas para evitar el Error 400
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    # CONFIGURACIÓN DE SALTO AUTOMÁTICO (Failover)
    # Si el servidor da error o está apagado, salta al siguiente de inmediato
    proxy_next_upstream error timeout http_500 http_502 http_503 http_504;

    # Tiempos de espera cortos para que el usuario no note el servidor caído
    proxy_connect_timeout 2s;
    proxy_read_timeout 10s;
    proxy_send_timeout 10s;
}
```

Reenvío de tráfico:

proxy_pass http://servidores_web;: Envía todo el tráfico que llega a / hacia el grupo de servidores que definiste arriba.

Cabeceras (Headers):

Estas líneas son para que los servidores finales sepan quién es el cliente real, y no piensen que todo el tráfico viene solo del proxy:

Host $http_host: Mantiene el nombre del dominio original.

X-Real-IP y X-Forwarded-For: Pasan la dirección IP real del usuario que visita la web.

X-Forwarded-Proto: Indica si el usuario original usó HTTP o HTTPS.

4. Failover:

```
    # CONFIGURACIÓN DE SALTO AUTOMÁTICO (Failover)
    # Si el servidor da error o está apagado, salta al siguiente de inmediato
    proxy_next_upstream error timeout http_500 http_502 http_503 http_504;

    # Tiempos de espera cortos para que el usuario no note el servidor caído
    proxy_connect_timeout 2s;
    proxy_read_timeout 10s;
    proxy_send_timeout 10s;
}
```

Esto sirve para que el usuario no vean errores:

proxy_next_upstream: Le dice a Nginx que si el servidor al que se intentase conectar da error 500, 502, 503, 504 o timeout, pasa la petición inmediatamente al siguiente servidor disponible".

proxy_connect_timeout 2s;: Si el servidor final no responde en 2 segundos, Nginx deja de esperar y busca al otro.

proxy_read/send_timeout 10s;: Límites de tiempo para enviar y recibir datos antes de cortar la conexión por lentitud.