

NGINX Reverse Proxy e Granja de Servidores IPv6 Only: Conectividade Web Eficiente

Autor: Alejandro Acosta

Coordenação e revisão: Guillermo Cicileo, Carlos MArtínez

Edição: Área de Comunicações

Área: Área de Tecnologia

Julho 2023

Introdução	2
O que é um reverse proxy?	2
O que é um servidor proxy?.....	2
Quais os benefícios de um Reverse Proxy?	3
Topologia que usaremos	4
O que vamos conseguir hoje?	4
Mãos à obra	5
Sobre os logs	7
Conclusões	8
Referências.....	8

Introdução

Neste trabalho apresentaremos uma forma muito simples de oferecer acesso Web Dual Stack a uma granja de servidores IPv6 Only, utilizando NGINX. Com o crescimento contínuo da rede de adoção gradual do protocolo IPv6, é essencial garantir conectividade e acessibilidade para os clientes que utilizam tanto IPv4 como IPv6.

Explicaremos como configurar NGINX para a admissão de acesso web Dual Stack; veremos como configurar NGINX como um proxy inverso que escuta tanto em endereços IPv4 quanto em IPv6 e como direcionar corretamente as solicitações entrantes aos servidores backend que possuem apenas endereços IPv6. Por sinal, o que estudaremos no seguinte artigo é um passo importante, entre muitos outros benefícios, para conseguir a tão desejada economia de endereços IPv4.

O que é um reverse proxy?

Cloudflare define em [1] um Servidor Proxy Inverso ou Reverso como:

“Um proxy inverso é um servidor situado na frente dos servidores webs que reenvia as solicitações do cliente (por exemplo, o navegador web) a esses servidores webs. Os proxys inversos costumam implementar-se para ajudar a aumentar a segurança, o rendimento e a fiabilidade. Para entender melhor o funcionamento de um proxy inverso, bem como as vantagens que pode acarretar, definiremos primeiro o que é um servidor proxy.”

O que é um servidor proxy?

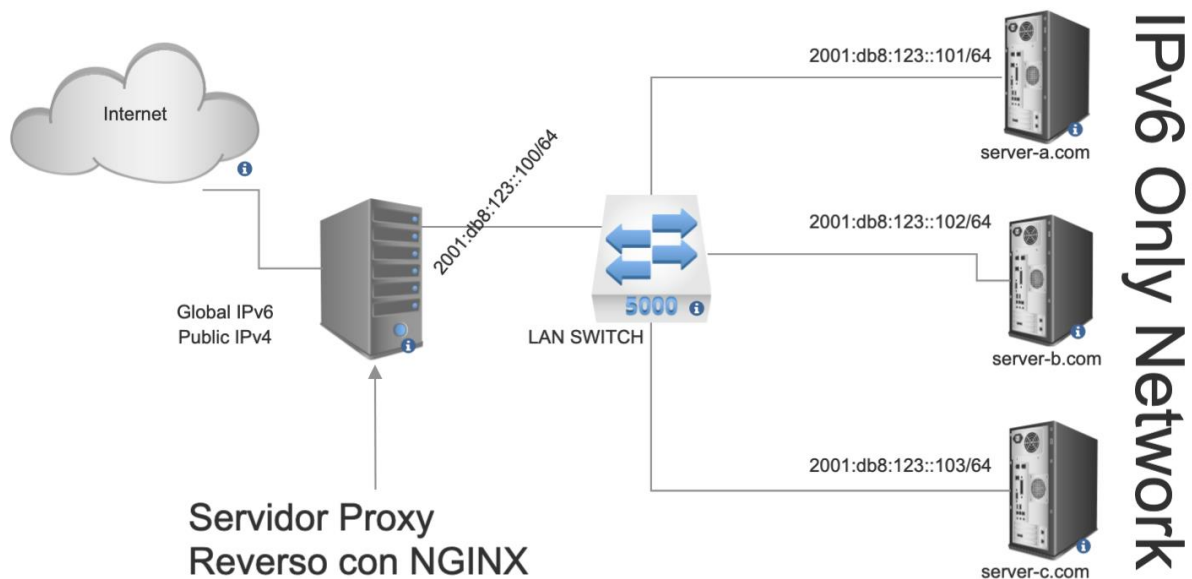
Novamente Cloudflare define em [1] um Servidor Proxy como:

“Um proxy de reenvio, com frequência conhecido como proxy, servidor proxy ou proxy web, é um servidor situado na frente de um grupo de máquinas clientes. Quando esses ordenadores realizam solicitações a sites e serviços na Internet, o servidor proxy intercepta esses pedidos e se comunica com os servidores web em nome desses clientes, como um intermediário.”

Quais os benefícios de um Reverse Proxy?

- Oferece IPv4 ou IPv6 transparente a clientes provenientes da Internet, servidos de uma granja de servidores IPv6 Only (neste ponto focaremos).
- **Escalabilidade:** Ao utilizar um proxy inverso, é possível adicionar ou excluir servidores backend, conforme for necessário, sem atingir os usuários finais. Isso facilita a escalabilidade horizontal dos aplicativos, permitindo lidar com um número maior de solicitações e usuários em simultâneo.
- **Cache de conteúdo estático:** NGINX pode armazenar em cache conteúdo estático como: imagens, arquivos CSS e JavaScript, reduzindo o carregamento nos servidores backend e acelerando a entrega de conteúdo aos usuários. Isso melhora o tempo de carregamento das páginas e reduz a largura de banda necessária.
- **Segurança:** NGINX age como um ponto de entrada ao aplicativo, proporcionando uma camada adicional de segurança. Pode realizar funções como filtragem de solicitações, prevenção de ataques DDoS, proteção contra injeções SQL e autenticação de clientes. Além disso, NGINX pode habilitar o uso de SSL/TLS para cifrar a comunicação entre os clientes e o servidor backend.
- **Consolidação de serviços:** NGINX pode agir como um ponto de entrada único para vários serviços backend. Isto simplifica a infraestrutura ao consolidar múltiplos serviços em um só servidor, facilitando a administração e a manutenção.
- **Melhoria do rendimento:** NGINX está desenhado para ser leve e eficiente no uso de recursos. Sua arquitetura otimizada e sua capacidade para lidar com grandes quantidades de conexões simultâneas o tornam uma opção popular para aprimorar o rendimento dos aplicativos webs.
- **Balanceamento de carga:** Um proxy reverso como NGINX pode distribuir o tráfego que entra através de vários servidores backend. Isto ajuda a equilibrar a carga de trabalho e garante que nenhum servidor esteja sobrecarregado, fazendo com que o rendimento e a capacidade de resposta do aplicativo melhorem.

Topologia que usaremos



O que vamos conseguir hoje?

O servidor na borda (Servidor Proxy Reverso) será capaz de receber pedidos HTTP em IPv4 e IPv6 e, dependendo do website que se deseja visitar (domínio), reenviará a consulta ao servidor correto. No exemplo atual ocorrerá o seguinte:

O Cliente visita	Pedido enviado a:
server-a.com	→ 2001:db8:123::1
server-b.com	→ 2001:db8:123::2
server-c.com	→ 2001:db8:123::3
server-a.com	→ 2001:db8:123::101
server-b.com	→ 2001:db8:123::102
server-c.com	→ 2001:db8:123::103

Pré-requisitos

- Linux com Nginx no Servidor Proxy Reverso
- Acesso super usuário
- Servidor Web em cada um dos servidores da granja
- Conectividade na Internet IPv4 e IPv6
- Conectividade interna no IPv6

Mãos à obra

- 1) Instalar nginx em todos os servidores

```
#apt update
#apt install nginx
```

- 2) Criar os websites no Proxy Reverso NGINX

Arquivo /etc/nginx/sites-available/server-a.com

```
server {
    listen 80;
    listen [::]:80;

    server_name server-a.com;
    location / {
        proxy_pass http://[2001:db8:123::101];
    }
}
```

Arquivo /etc/nginx/sites-available/server-b.com

```
server {
    listen 80;
    listen [::]:80;

    server_name server-b.com;
    location / {
        proxy_pass http://[2001:db8:123::102];
    }
}
```

Arquivo /etc/nginx/sites-available/server-c.com

```
server {
    listen 80;
    listen [::]:80;

    server_name server-c.com;
    location / {
        proxy_pass http://[2001:db8:123::103];
    }
}
```

3) Criar links simbólicos para habilitar os sites configurados:

```
root@ProxyReverseSRV:/etc/nginx/sites-enabled# ln -s  
/etc/nginx/sites-available/server-a.com /etc/nginx/sites-  
enabled/server-a.com
```

```
root@ProxyReverseSRV:/etc/nginx/sites-enabled# ln -s  
/etc/nginx/sites-available/server-b.com /etc/nginx/sites-  
enabled/server-b.com
```

```
root@ProxyReverseSRV:/etc/nginx/sites-enabled# ln -s  
/etc/nginx/sites-available/server-c.com /etc/nginx/sites-  
enabled/server-c.com
```

4) Lembrem de reiniciar nginx:

```
$sudo systemctl restart nginx
```

Sobre os logs

Os registros de conexão (logs) são de extrema importância para qualquer empresa e ISP que deseje realizar alguma revisão das conexões entrantes.

O que acontece é que NGINX por defeito utilizará seu próprio endereço IP quando realizar as conexões salientes, o que acarretará como consequência a perda do endereço do cliente que originou a solicitação HTTP. Porém, não se preocupem, NGINX tem a solução que se chama `proxy_set_header` e a configuração se divide no servidor final e no servidor Proxy Reverso.

No Servidor Proxy Reverso, arquivo do website.

Exemplo do nginx reverse proxy que permite conservar o endereço # e o porto original do cliente

```
location /examples {
    proxy_pass http://[2001:db8:123::103];
    proxy_buffering off;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Port $server_port;
    proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
}
```

No servidor final do arquivo: `etc/nginx/nginx.conf` adicionar na seção `http` o seguinte:

```
set_real_ip_from 2001:db8:123::100; #substituir o
endereço IP pelo Proxy
real_ip_header X-Forwarded-For;
real_ip_recursive on;
```

Exemplo:

```
http {
    ...
    set_real_ip_from 2001:db8:123::100;
    real_ip_header X-Forwarded-For;
    real_ip_recursive on;
    ...
}
```

Após estas configurações o servidor final confiará na cabeceira chamada `X-Forwarded-For` que provenha do IP `2001:db8:123::100` e em seus registros (`/var/log/nginx/access.log`) será possível visualizar o endereço de origem do cliente.

Conclusões

Dá para perceber que com o desenho proposto podemos administrar uma granja de servidores web 100% IPv6 Only com acesso ao mundo, tanto IPv4 como IPv6, de uma forma muito simples, escalável e eficiente. Isso traz consigo muitos benefícios, tais como: administrar apenas um stack TCP/IP, simplicidade, segurança e inclusive economizar endereços IPv4.

Referências

- [1] <https://www.cloudflare.com/es-es/learning/cdn/glossary/reverse-proxy/>
- <https://www.digitalocean.com/community/tutorials/how-to-configure-nginx-as-a-reverse-proxy-on-ubuntu-22-04>
- Arquivos de configuração de todo o projeto no Github do LACNIC:
https://github.com/LACNIC/BlogPostHelpFiles/tree/main/2023_Ofreciendo_conectividad_ad_Dual_Stack_a_servidores_Web_en_una_granja_de_servidores_100_IPv6_Only