

# Atividade 08 - ASR Telemática

## PROXY REVERSO:

Um proxy reverso é um servidor que atua como intermediário entre os clientes (usuários finais) e os servidores backend (aplicações ou serviços internos). Ele recebe solicitações dos clientes, as processa e as encaminha para o servidor apropriado, retornando a resposta ao cliente. Diferente de um proxy direto (que protege o cliente), o proxy reverso protege e otimiza o acesso aos servidores backend.

## Principais conceitos:

- Roteamento de Solicitações: O proxy reverso decide para qual backend enviar a solicitação com base em regras (ex.: URL, cabeçalhos, balanceamento de carga).
- Balanceamento de Carga: Distribui o tráfego entre múltiplos servidores backend para evitar sobrecarga, melhorando disponibilidade e desempenho.
- Cache: Armazena respostas estáticas ou dinâmicas para reduzir a carga nos backends e acelerar respostas.
- Terminação SSL/TLS: O proxy pode lidar com criptografia HTTPS, descriptografando solicitações e criptografando respostas, aliviando os backends.
- Segurança: Pode filtrar solicitações maliciosas, ocultar a arquitetura interna e adicionar autenticação.
- Compressão e Otimização: Comprime respostas e otimiza conteúdo para reduzir latência.

## NGINX:

O NGINX é uma ferramenta popular para implementar proxies reversos devido à sua eficiência, baixo consumo de recursos e suporte a alta concorrência. Ele usa um modelo assíncrono baseado em eventos, permitindo lidar com milhares de conexões simultâneas.

A configuração do NGINX é baseada em arquivos de texto simples, geralmente localizados em `/etc/nginx` (em sistemas Linux). O arquivo principal é `nginx.conf`, que define diretivas globais, módulos e blocos de servidor.

## Arquivos Principais:

- **nginx.conf:** Arquivo de configuração global. Inclui diretivas como `worker_processes`, `events` e blocos `http` ou `stream`.
- **sites-available/ e sites-enabled/:** Em distribuições como Ubuntu, configurações específicas de sites são colocadas em `sites-available` e habilitadas via link simbólico em `sites-enabled`.
- **upstream:** Bloco para definir grupos de servidores backend (ex.: para balanceamento de carga).

## Implementando uma Solução de Proxy Reverso:

2 Containers:

Backend Simples: Um container com Apache HTTP Server (`httpd`) como backend.

Proxy Reverso: Um container NGINX configurado para rotear para o backend

Criando uma rede Docker para conectar os containers:

*`docker network create mynetwork.`*

Executando o backend:

*`docker run -d --name backend --network mynetwork -p 8080:80 httpd.`*

Isso inicia um servidor Apache na porta 8080 interna, servindo uma página padrão.

Crie um arquivo de configuração NGINX localmente no PWD (use o editor online ou vi).

```
1 events {
2     worker_connections 1024;
3 }
4
5 http {
6     upstream app_backend {
7         server backend:80;
8     }
9
10    server {
11        listen 80;
12        server_name localhost;
13
14        location / {
15            proxy_pass http://app_backend;
16            proxy_set_header Host $host;
17            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
18            proxy_set_header X-Real-IP $remote_addr;
19        }
20    }
21 }
```

- Executando o NGINX:

```
docker run -d --name proxy --network mynetwork -p 80:80 -v  
$(pwd)/nginx.conf:/etc/nginx/nginx.conf nginx.
```

## Testando o Uso do Proxy Reverso:

### 1) Acesso direto ao Backend:

```
[node1] (local) root@192.168.0.24 ~  
$ docker ps  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                NAMES  
686136afc5cd   nginx     "/docker-entrypoint...." 8 seconds ago  Up 8 seconds  0.0.0.0:80->80/tcp    proxy  
6d2ee646c4a5   httpd     "httpd-foreground"        17 minutes ago Up 17 minutes  0.0.0.0:8080->80/tcp   backend  
[node1] (local) root@192.168.0.24 ~  
$ curl http://localhost:80  
<!DOCTYPE html>  
<html>  
<head>  
<title>Welcome to nginx!</title>  
<style>  
html { color-scheme: light dark; }  
body { width: 35em; margin: 0 auto;  
font-family: Tahoma, Verdana, Arial, sans-serif; }  
</style>  
</head>  
<body>  
<h1>Welcome to nginx!</h1>  
<p>If you see this page, the nginx web server is successfully installed and  
working. Further configuration is required.</p>  
  
<p>For online documentation and support please refer to  
<a href="http://nginx.org/">nginx.org</a>.<br/>  
Commercial support is available at  
<a href="http://nginx.com/">nginx.com</a>.</p>  
  
<p><em>Thank you for using nginx.</em></p>  
</body>  
</html>  
[node1] (local) root@192.168.0.24 ~  
$
```

### 2) Teste via Browser:

No PWD, clique em "Open Port" ao lado da porta 80 do container "proxy".

Uma nova aba abrirá com uma URL como

<http://ip-10-0-0-1-80.direct.labs.play-with-docker.com>.

