

Atividade-08

1) Explique os conceitos inerentes à implementação de um proxy reverso implementado com o NGINX.

O NGINX atuando como proxy reverso funciona como um intermediário entre os clientes (usuários na internet) e os servidores de aplicação (backend). Ele possui um Ponto de Entrada Único onde o cliente faz a requisição para o NGINX, e este decide para qual servidor interno encaminhar o tráfego. Sua segurança esconde a identidade e a estrutura da rede interna (os endereços IP reais dos servidores de aplicação não ficam expostos). Possui balanceamento de carga podendo distribuir as requisições entre vários servidores para evitar sobrecarga. E possui uma Terminação SSL/TLS, onde o NGINX pode lidar com a criptografia, aliviando o processamento do servidor de aplicação.

2) Descreva como funciona a configuração e arquivos envolvidos.

No NGINX em distribuições Linux comuns por exemplo, os arquivos principais são:

1. /etc/nginx/nginx.conf: O arquivo de configuração principal. Ele carrega as configurações globais.
2. /etc/nginx/conf.d/: Diretório onde geralmente colocamos nossas configurações personalizadas
- .
3. Diretiva proxy_pass: É a alma do proxy reverso. Ela diz ao NGINX para onde enviar a requisição recebida.

Implemente uma solução de proxy reverso usando a máquina virtual ou o play with docker. Faça print dos arquivos envolvendo a descrição dos conceitos e os arquivos de configuração. Faça print dos testes que demonstram o uso do Proxy reverso.

print 1

print 2

```
01:30:42
d50utlc6_d50v7tol2o9000fdk3k0

CLOSE SESSION

IP: 192.168.0.24 OPEN PORT: 80
Memory: 5.04% (201.7MiB / 3.906GiB) CPU: 0.17%
Instances: SSH ssh ip172-18-0-14-d50utlc69qj000bt03og@direct.labs.play [ ] + ADD NEW INSTANCE

192.168.0.24
node1

DELETE EDITOR

< />
> ZOF
[node1] (local) root@192.168.0.24 ~
$ docker run -d --name nginx -p 80:80 --network rede-proxy -v $(pwd)/nginx.conf:/etc/nginx/nginx.conf:ro nginx:alpine
Unbuilt image: nginx:alpine
alpine: Pulling from library/nginx
014e56e6e13961: Already exists
df5d290a5c25: Full complete
5d5a0a0f3a7a: Full complete
abdec0946203: Full complete
51c30493937c: Full complete
ad5b5d5aa02cf: Full complete
4ec5a2a3a20e: Full complete
134bc697c201: Full complete
Digest: sha256:052b75a72ff690f33ddeb51c7e08d9696a0447a133eb2b99cc905d9188cb2b
Status: Downloaded newer image for nginx:alpine
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
79534ce6e92 nginx:alpine "/usr/local/bin/docker-entrypoint -" 23 seconds ago Up 24 seconds 0.0.0.0:80->80/tcp nginx
79534ce6e92 httpd:foreground "httpd-foreground" 1 minute ago Up 1 minute 80/tcp meu-backend
[node1] (local) root@192.168.0.24 ~
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
79534ce6e92 nginx:alpine "/usr/local/bin/docker-entrypoint -" 23 seconds ago Up 24 seconds 0.0.0.0:80->80/tcp nginx
79534ce6e92 httpd:foreground "httpd-foreground" 1 minute ago Up 1 minute 80/tcp meu-backend
[node1] (local) root@192.168.0.24 ~
$ curl localhost
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 13
Date: Mon, 10 Jun 2019 01:30:42 GMT
Server: Apache/2.4.29 (Debian)
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>It works! Apache httpd</title>
</head>
<body>
<p>It works!</p>
</body>
</html>
[node1] (local) root@192.168.0.24 ~
```

print 3

print 4

It works!

