

```
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
Terminal - icaro@icaro-Virtu

A processar 'triggers' para man-db (2.10.2-1) ...
icaro@icaro-VirtualBox:~$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
icaro@icaro-VirtualBox:~$ sudo docker run -d --name nginx-server -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
9e3ea8720c6d: Pull complete
bf36b6466679: Pull complete
15a97cf85bb8: Pull complete
9c2d6be5a61d: Pull complete
6b7e4a5c7c7a: Pull complete
8db4caa19df8: Pull complete
Digest: sha256:480868e8c8c797794257e2abd88d0f9a8809b2fe956cbfbc05dcc0bca1f7cd43
Status: Downloaded newer image for nginx:latest
5189dd415f0a669403dbf27f93c37b6cd901b969d6dab3a5f7d4d43de1bd56f
icaro@icaro-VirtualBox:~$ docker volume create nginx-data
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/volumes/create": dial unix:///var/run/docker.sock: connect: permission denied
icaro@icaro-VirtualBox:~$ sudo docker volume create nginx-data
nginx-data
icaro@icaro-VirtualBox:~$ sudo docker volume inspect nginx-data
[
  {
    "CreatedAt": "2023-05-17T21:00:36-03:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/nginx-data/_data",
    "Name": "nginx-data",
    "Options": {},
    "Scope": "local"
  }
]
```

```
icaro@icaro-VirtualBox:~$ sudo ln -s /var/lib/docker/volumes/nginx-data/_data /nginx
icaro@icaro-VirtualBox:~$ sudo docker run -d --name nginx-server -p 80:80 -v nginx-data:/usr/share/nginx/html nginx
docker: Error response from daemon: Conflict. The container name "/nginx-server" is already in use by container "5189dd415f0a669403dbf27f93c37b6cd901b969d6dab3a5f7d4d43de1bd56f". You can not reuse a container name that is already in use by another container.
See 'docker run --help'.
icaro@icaro-VirtualBox:~$ sudo docker run -d --name nginx-server -p 80:80 -v nginx-data:/usr/share/nginx/html nginx
docker: Error response from daemon: Conflict. The container name "/nginx-server" is already in use by container "5189dd415f0a669403dbf27f93c37b6cd901b969d6dab3a5f7d4d43de1bd56f". You can not reuse a container name that is already in use by another container.
See 'docker run --help'.
icaro@icaro-VirtualBox:~$ sudo ls /var/lib/docker/volumes/nginx-data/_data
icaro@icaro-VirtualBox:~$ sudo 50x.html index.html
sudo: 50x.html: comando não encontrado
icaro@icaro-VirtualBox:~$ 50x.html index.html
50x.html: comando não encontrado
icaro@icaro-VirtualBox:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5189dd415f0a nginx "/docker-entrypoint..." 4 minutes ago Up 4 minutes 0.0.0.0:80->80/tcp, :::80->80/tcp nginx-server
icaro@icaro-VirtualBox:~$ sudo docker ps -a -f id=177e8effa98fdf8fcc87d377203106d80502cfce10ef7cc619af859577841f56
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5189dd415f0a nginx "/docker-entrypoint..." 4 minutes ago Up 4 minutes 0.0.0.0:80->80/tcp, :::80->80/tcp nginx-server
icaro@icaro-VirtualBox:~$ sudo docker container stop 177e8effa98fdf8fcc87d377203106d80502cfce10ef7cc619af859577841f56
Error response from daemon: No such container: 177e8effa98fdf8fcc87d377203106d80502cfce10ef7cc619af859577841f56
icaro@icaro-VirtualBox:~$ sudo docker container start nginx-server
nginx-server
icaro@icaro-VirtualBox:~$ sudo docker container start 177e8effa98fdf8fcc87d377203106d80502cfce10ef7cc619af859577841f56
Error response from daemon: No such container: 177e8effa98fdf8fcc87d377203106d80502cfce10ef7cc619af859577841f56
Error: failed to start containers: 177e8effa98fdf8fcc87d377203106d80502cfce10ef7cc619af859577841f56
icaro@icaro-VirtualBox:~$ sudo docker container start nginx-server
```

```
icaro@icaro-VirtualBox:~$ sudo docker logs nginx-server
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/05/17 23:59:45 [notice] 1#1: using the "epoll" event method
2023/05/17 23:59:45 [notice] 1#1: nginx/1.23.4
2023/05/17 23:59:45 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/05/17 23:59:45 [notice] 1#1: OS: Linux 5.19.0-40-generic
2023/05/17 23:59:45 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/05/17 23:59:45 [notice] 1#1: start worker processes
2023/05/17 23:59:45 [notice] 1#1: start worker process 29
2023/05/18 00:05:32 [notice] 1#1: signal 3 (SIGQUIT) received, shutting down
2023/05/18 00:05:32 [notice] 29#29: gracefully shutting down
2023/05/18 00:05:32 [notice] 29#29: exiting
2023/05/18 00:05:32 [notice] 29#29: exit
2023/05/18 00:05:32 [notice] 1#1: signal 17 (SIGCHLD) received from 29
2023/05/18 00:05:32 [notice] 1#1: worker process 29 exited with code 0
2023/05/18 00:05:32 [notice] 1#1: exit
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: IPv6 listen already enabled
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/05/18 00:06:11 [notice] 1#1: using the "epoll" event method
2023/05/18 00:06:11 [notice] 1#1: nginx/1.23.4
2023/05/18 00:06:11 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/05/18 00:06:11 [notice] 1#1: OS: Linux 5.19.0-40-generic
2023/05/18 00:06:11 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/05/18 00:06:11 [notice] 1#1: start worker processes
2023/05/18 00:06:11 [notice] 1#1: start worker process 22
icaro@icaro-VirtualBox:~$
```

Faça as seguintes perguntas ao Chat GPT:

Eu: O que é um proxy reverso?

Chat GPT: Um proxy reverso é um tipo de servidor intermediário que atua como uma interface entre os clientes na Internet e os servidores de destino, redirecionando as solicitações dos clientes para os servidores adequados. Ao contrário de um proxy normal, que age em nome dos clientes para acessar recursos externos, o proxy reverso age em nome dos servidores para fornecer serviços aos clientes.

O funcionamento básico de um proxy reverso envolve receber as solicitações dos clientes e, em seguida, encaminhá-las para os servidores de destino apropriados. Isso pode ser usado para diversos fins, como melhorar o desempenho e a segurança de um servidor web, balanceamento de carga, cache de conteúdo, filtragem de tráfego e até mesmo ocultar a localização dos servidores internos.

Ao utilizar um proxy reverso, os clientes se comunicam diretamente com o proxy, que, por sua vez, encaminha as solicitações para os servidores apropriados. Isso permite que os servidores permaneçam ocultos atrás do proxy, aumentando a segurança e reduzindo a exposição direta à Internet.

Em resumo, um proxy reverso é um servidor intermediário que gerencia as solicitações dos clientes e as redireciona para os servidores de destino corretos, melhorando o desempenho, a segurança e a eficiência das conexões de rede.

Como se configura um proxy reverso no NGINX?

Para configurar um proxy reverso no NGINX, você precisa seguir os seguintes passos:

1 - Instale o NGINX: Comece instalando o NGINX no seu sistema. Os comandos de instalação podem variar dependendo do sistema operacional que você está usando.

2 - Edite o arquivo de configuração do NGINX: Abra o arquivo de configuração principal do NGINX. O caminho típico é /etc/nginx/nginx.conf. Use um editor de texto para fazer as alterações necessárias.

3 - Defina o servidor upstream: Dentro do bloco http, você precisa definir o servidor de destino (upstream) para o qual o proxy reverso encaminhará as solicitações.

Aqui está um exemplo:

```
http {  
    upstream backend {  
        server backend-server1;  
        server backend-server2;  
    }  
}
```

Substitua backend-server1 e backend-server2 pelos endereços IP ou nomes de domínio dos seus servidores de destino.

4 - Configure a diretiva do servidor: Dentro do bloco server, você precisa configurar as diretivas para o servidor proxy reverso. Aqui está um exemplo básico:

```
server {  
    listen 80;  
    server_name seu_dominio.com;  
  
    location / {  
        proxy_pass http://backend;  
    }  
}
```

Certifique-se de substituir seu_dominio.com pelo seu próprio domínio. A diretiva proxy_pass define o servidor upstream que você definiu anteriormente.

5 - Salve as alterações e reinicie o NGINX: Após fazer as alterações no arquivo de configuração, salve-o e reinicie o serviço NGINX para que as alterações entrem em vigor. O comando para reiniciar pode variar dependendo do seu sistema operacional, mas geralmente é algo como `sudo service nginx restart` ou `sudo systemctl restart nginx`.

Após reiniciar o NGINX, seu proxy reverso estará configurado e pronto para redirecionar as solicitações dos clientes para os servidores de destino especificados. Certifique-se de ajustar as configurações conforme necessário, como portas, endereços IP e nomes de domínio, para corresponder à sua configuração específica.