



viniciusgjusta

Search by repository name



All Content

Create repository

viniciusgjusta / nginx-container

Contains: Image | Last pushed: a few seconds ago



Inactive



0



0



Public



Terminal

mai 16 21:27



vinicius@vinicius-VirtualBox: ~



```
vinicius@vinicius-VirtualBox:~$ sudo docker pull nginx
```

```
[sudo] password for vinicius:
```

```
Using default tag: latest
```

```
latest: Pulling from library/nginx
```

```
9e3ea8720c6d: Pull complete
```

```
bf36b6466679: Pull complete
```

```
15a97cf85bb8: Pull complete
```

```
9c2d6be5a61d: Pull complete
```

```
6b7e4a5c7c7a: Pull complete
```

```
8db4caa19df8: Pull complete
```

```
Digest: sha256:480868e8c8c797794257e2abd88d0f9a8809b2fe956cbfbc05dcc0b  
ca1f7cd43
```

```
Status: Downloaded newer image for nginx:latest
```

```
docker.io/library/nginx:latest
```



Home

```
Terminal mai 16 21:28
```

+

vinicius@vinicius-VirtualBox: ~

Q

≡

-

□

×

```
web
vinicius@vinicius-VirtualBox:~$ sudo docker run -d --name nginx-server
-p 80:80 nginx
255238edc9b48cf94ff1c12333c602af0ca2ad17184bd3617bcc6263717cef71
vinicius@vinicius-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS      NAMES
255238edc9b4   nginx    "/docker-entrypoint.…"   18 seconds    Up         nginx-ser
17 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp
a2f16b7a2173   nginx    "/docker-entrypoint.…"   5 minutes ago Exited (0)   Exited
4 minutes ago                                focused_j
epsen
d6636de7e847   nginx    "/docker-entrypoint.…"   12 minutes ago Exited (0)   Exited
11 minutes ago                                funny_kil
by
5c4f9cc54227   ubuntu   "/bin/bash"              3 days ago    Exited (137) Exited
3 days ago                                charming_
brown
vinicius@vinicius-VirtualBox:~$ sudo docker rm a2f16b7a2173
```

Home

```
Terminal mai 16 21:28
vinicius@vinicius-VirtualBox: ~
Exited (137) 3 days ago charmin
g_brown
vinicius@vinicius-VirtualBox:~$ sudo docker commit -m "initialized ngi
nx container" -a "viniciusgjusta" 255238edc9b4 viniciusgjusta/nginx-co
ntainer
sha256:2c74e896415c187561705d4aaef483b4a485586f2357cb24430378e16223ef8
b
vinicius@vinicius-VirtualBox:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS      PORTS          NAMES
255238edc9b4   nginx    "/docker-entrypoint.…"   6 minutes ago   Up 6 min
0.0.0.0:80->80/tcp, :::80->80/tcp   nginx-server
5c4f9cc54227   ubuntu   "/bin/bash"             3 days ago      Exited (137) 3 days ago   charming_bro
wn
vinicius@vinicius-VirtualBox:~$ sudo docker images
REPOSITORY              TAG          IMAGE ID          CREATED
SIZE
viniciusgjusta/nginx-container  latest      2c74e896415c     11 seconds a
go 142MB
```

```
Terminal mai 16 21:29
vinicius@vinicius-VirtualBox: ~
vinicius@vinicius-VirtualBox:~$ sudo docker push viniciusgjusta/nginx-
container
Using default tag: latest
The push refers to repository [docker.io/viniciusgjusta/nginx-containe
r]
dfe6d166edda: Pushed
1040838fe30e: Mounted from library/nginx
93ee76f39c97: Mounted from library/nginx
5684be535bf1: Mounted from library/nginx
6bc8ae8fb3cf: Mounted from library/nginx
a29cc9587af6: Mounted from library/nginx
8553b91047da: Mounted from library/nginx
latest: digest: sha256:d0d7e2fafeec183c259aeff490e8a7f4df457c59ee4e883
```



Um proxy reverso, também conhecido como reverse proxy, é um servidor intermediário que atua como uma camada de frente para um ou mais servidores web. Ao contrário de um proxy convencional, que age como intermediário entre um cliente e um servidor, um proxy reverso fica entre os clientes e os servidores, agindo em nome dos servidores web.



Quando um cliente faz uma solicitação para um determinado recurso, como uma página da web, ele envia a solicitação para o proxy reverso. O proxy reverso, por sua vez, encaminha essa solicitação para o servidor web apropriado. O servidor web processa a solicitação e envia a resposta de volta ao proxy reverso, que então encaminha a resposta ao cliente.

Existem várias razões para usar um proxy reverso. Algumas delas incluem:

1. Balanceamento de carga: Um proxy reverso pode distribuir o tráfego entre vários servidores web, ajudando a melhorar o desempenho e a capacidade de resposta do sistema.

🔄 Regenerate response

2. Cache: O proxy reverso pode armazenar em cache as respostas dos servidores web para

2. Cache: O proxy reverso pode armazenar em cache as respostas dos servidores web para solicitações comuns. Isso pode reduzir a carga nos servidores e melhorar o tempo de resposta para solicitações subsequentes.
3. Segurança: O proxy reverso atua como uma barreira entre os clientes e os servidores web, ocultando detalhes da infraestrutura interna. Isso pode ajudar a proteger os servidores contra ataques diretos.
4. SSL/TLS termination: O proxy reverso pode lidar com o tráfego criptografado SSL/TLS, fazendo a descriptografia e a criptografia necessárias antes de encaminhar as solicitações para os servidores web. Isso pode aliviar a carga nos servidores e simplificar a configuração do SSL/TLS.
5. Roteamento baseado em conteúdo: O proxy reverso pode encaminhar solicitações para servidores diferentes com base no conteúdo da solicitação. Por exemplo, ele pode direcionar solicitações para diferentes servidores com base no domínio solicitado.

Em resumo, um proxy reverso é um servidor intermediário que gerencia e encaminha solicitações dos clientes para servidores web, fornecendo recursos como balanceamento de carga, cache, segurança e roteame

🔄 Regenerate response

VI

## Como configurar um proxy reverso no NGINX?



Para configurar um proxy reverso no NGINX, você precisa seguir as seguintes etapas:



1. Instalação do NGINX: Primeiro, certifique-se de ter o NGINX instalado em seu sistema. Você pode instalá-lo usando os comandos apropriados para o seu sistema operacional.
2. Configuração do arquivo de configuração: O arquivo de configuração principal do NGINX é geralmente chamado de `nginx.conf` e está localizado em um diretório específico, dependendo do sistema operacional. Abra o arquivo de configuração em um editor de texto.
3. Definir o bloco de servidor: Dentro do arquivo de configuração, você precisará definir um bloco de servidor para configurar o proxy reverso. Um exemplo básico é:

perl

Copy code

```
server {  
    listen 80;
```

Regenerate response

perl

Copy code

```
server {  
    listen 80;  
    server_name exemplo.com;  
  
    location / {  
        proxy_pass http://endereco-do-servidor;  
    }  
}
```

- `listen`: Especifica a porta na qual o NGINX vai escutar as solicitações (no exemplo, a porta 80).
- `server_name`: Define o nome do servidor (no exemplo, "exemplo.com").
- `location`: Especifica a localização a ser direcionada para o proxy reverso.
- `proxy_pass`: Indica o endereço do servidor para o qual as solicitações serão encaminhadas.


Regenerate response

4. Configurar outras opções: Além disso, você pode configurar outras opções,

4. Configurar outras opções: Além do `proxy_pass`, você pode configurar outras opções, como cache, SSL/TLS, autenticação, entre outros, dentro do bloco de servidor ou do bloco de localização. Essas opções dependem dos requisitos específicos do seu ambiente.
5. Salvar e fechar o arquivo de configuração.
6. Reiniciar o NGINX: Após fazer as alterações no arquivo de configuração, reinicie o serviço do NGINX para que as alterações entrem em vigor. O comando pode variar dependendo do seu sistema operacional, mas geralmente é algo como `sudo systemctl restart nginx`.

É importante ressaltar que essa é apenas uma configuração básica de proxy reverso. Você pode personalizar e adicionar mais opções conforme necessário, como balanceamento de carga, cache, autenticação, entre outros recursos oferecidos pelo NGINX.

Certifique-se de verificar a documentação oficial do NGINX para obter mais informações sobre a configuração do proxy reverso e para se familiarizar com as opções disponíveis.

 Regenerate response