

Tutorial: implantação de Servidor Web com NGINX em container docker no VirtualBox

Daiane Maria dos Santos Ribeiro¹, Nereu Vítor Pereira Lima²

¹Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano –
IFSertãoPE Campus Salgueiro

BR 232, km 508, sentido Recife – 56.000-000 – Salgueiro – PE – Brasil

{daiane.ribeiro@ifsertao-pe.edu.br, nereu.vitor@aluno.ifsertao-pe.edu.br}

Abstract. *This work's main objective is to present a tutorial on the installation and use of the NGINX web server, providing its functionalities and requirements necessary for its use in the Windows operating system.*

Resumo. *Este trabalho tem como objetivo principal apresentar um tutorial de instalação e uso do servidor web NGINX, fornecendo as suas funcionalidades e requisitos necessários para a utilização do mesmo no sistema operacional Windows.*

1. Introdução

Um servidor web é um sistema de software e hardware responsável por receber, interpretar e responder a requisições feitas por clientes via protocolo de transferência de hipertexto (HTTP, Hypertext Transfer Protocol) ou protocolo de transferência de hipertexto seguro (HTTPS, Hypertext Transfer Protocol Secure), geralmente entregando páginas web e serviços digitais (W3C, 2023).

Para ilustrar como um servidor web funciona nós iremos apresentar nesse trabalho um tutorial para instalação, configuração e utilização do servidor web NGINX em ambiente Windows.

A escolha desse servidor se deu a partir de uma pesquisa feita no chatGPT, por meio do seguinte prompt: “Analise os servidores Apache, LiteSpeed e NGINX usando 15 critérios de hardware e software em contexto de: 1- Servidor, 2- PC cliente, 3- SOC (Banana Pi, Raspberry Pi, Orange Pi ou Beagle). Em seguida, considere as características de hardware dos 3 servidores e forneça os componentes essenciais de cada um deles. Considere que o servidor será instalado numa máquina cliente Dell optiplex 3080 e forneça instrução de qual servidor é mais adequado usar nessa máquina”.

Considerando esse prompt, o chatGPT fez uma análise comparativa dos três servidores web (Apache, LiteSpeed e NGINX). A partir dessa análise percebemos que o servidor mais adequado para ser instalado em um computador de uso pessoal foi o NGINX. Ele apresenta algumas vantagens em relação aos outros, como por exemplo, excelente aproveitamento de CPU e RAM, ótimo para lidar com múltiplas conexões simultâneas,

muito seguro e eficiente, ideal para ambientes de produção com WordPress, Laravel, APIs REST etc, alto desempenho mesmo em máquinas mais modestas, entre outras.

A instalação do NGINX e seus recursos pode ser realizada através do site oficial do Nginx, onde contém a documentação para guiar e auxiliar em futuras dúvidas que possam surgir durante o seu uso. Além disso, no site, também, é possível encontrar as versões deste servidor destinadas à outros sistemas operacionais.

2. Passo a passo da instalação e uso do container e NGINX no VirtualBox

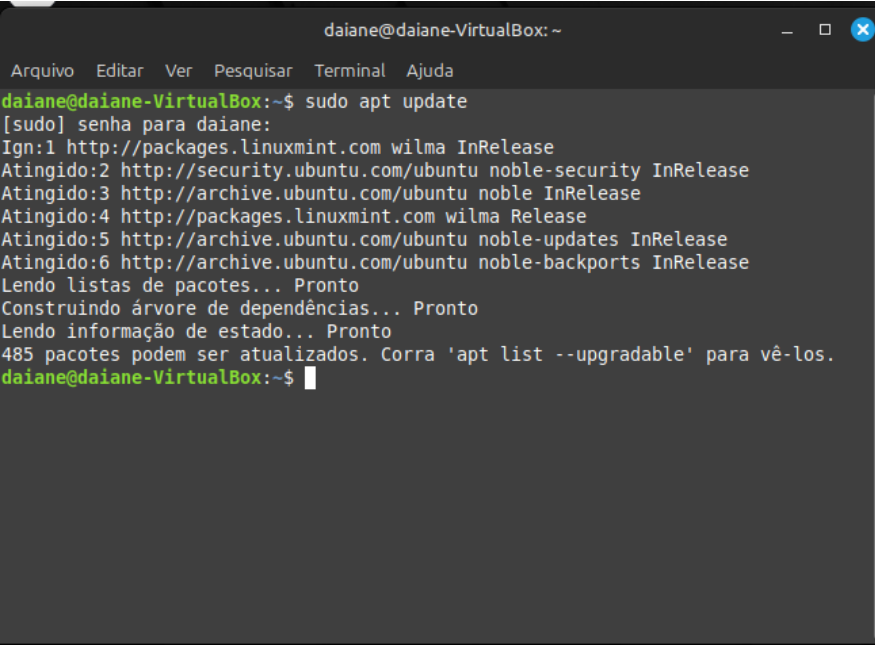
Nesta sessão iremos fornecer o passo a passo que realizamos desde a inicialização da Máquina Virtual (MV) até o uso do servidor NGINX dentro do container no VirtualBox.

2.1. Etapa 1 - Instalando o Docker na VM Linux

1. O VirtualBox e Ubuntu Server já estava instalado no notebook. Então, nós inicializamos a MV o logamos no Ubuntu.

2. Em seguida, abrimos o terminal e executamos os comandos:

- `sudo apt update` – Esse comando atualizou a lista de pacotes disponíveis nos repositórios do Ubuntu. Isso é importante, porque garante que o sistema saiba quais são as versões mais recentes dos programas antes de instalar qualquer coisa, como mostra a figura 1.



```
daiane@daiane-VirtualBox: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
daiane@daiane-VirtualBox:~$ sudo apt update  
[sudo] senha para daiane:  
Ign:1 http://packages.linuxmint.com wilma InRelease  
Atingido:2 http://security.ubuntu.com/ubuntu noble-security InRelease  
Atingido:3 http://archive.ubuntu.com/ubuntu noble InRelease  
Atingido:4 http://packages.linuxmint.com wilma Release  
Atingido:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease  
Atingido:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease  
Lendo listas de pacotes... Pronto  
Construindo árvore de dependências... Pronto  
Lendo informação de estado... Pronto  
485 pacotes podem ser atualizados. Corra 'apt list --upgradable' para vê-los.  
daiane@daiane-VirtualBox:~$
```

Figura 1. Atualização de listas de pacotes no Ubuntu.

- `sudo apt install -y docker.io` – Esse commando instalou o pacote `docker.io`, que é o Docker disponível nos repositórios do Ubuntu. Assim, o Docker foi instalado na MV, como ilustra a figura 2, permitindo a criação e gerenciamento de containers.

```
daiane@daiane-VirtualBox: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
Lendo listas de pacotes... Pronto  
Construindo árvore de dependências... Pronto  
Lendo informação de estado... Pronto  
485 pacotes podem ser atualizados. Corra 'apt list --upgradable' para vê-los.  
daiane@daiane-VirtualBox:~$ sudo apt install -y docker.io  
Lendo listas de pacotes... Pronto  
Construindo árvore de dependências... Pronto  
Lendo informação de estado... Pronto  
Os pacotes adicionais seguintes serão instalados:  
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan  
Pacotes sugeridos:  
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx  
  docker-compose-v2 docker-doc rinse git-daemon-run | git-daemon-sysvinit  
  git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn  
Os NOVOS pacotes a seguir serão instalados:  
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc  
  ubuntu-fan  
0 pacotes atualizados, 9 pacotes novos instalados, 0 a serem removidos e 485 não atualizados.  
É preciso baixar 83,7 MB de arquivos.  
Depois desta operação, 323 MB adicionais de espaço em disco serão usados.  
Obter:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65,6 kB]  
Obter:2 http://archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33,9 kB]  
Obter:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.2.5-0ubuntu1-24.04.1 [8.043 kB]  
Obter:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.27-0ubuntu1-24.04.1 [37,7 MB]  
Obter:5 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 27.5.1-0ubuntu3-24.04.2 [33,0 MB]  
Ign:5 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 27.5.1-0ubuntu3-24.04.2  
Obter:6 http://archive.ubuntu.com/ubuntu noble/main amd64 liberror-perl all 0.17029-2 [25,6 kB]  
Obter:7 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 git-man all 1:2.43.0-1ubuntu7.2 [1.100 kB]  
Obter:8 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 git amd64 1:2.43.0-1ubuntu7.2 [3.679 kB]  
Obter:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35,2 kB]  
Obter:5 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 27.5.1-0ubuntu3-24.04.2 [33,0 MB]  
86% [5 docker.io 20,7 MB/33,0 MB 63%]
```

Figura 2. Instalação do docker.io.

- `sudo systemctl start docker` – Esse comando iniciou o serviço do Docker imediatamente.
- `sudo systemctl enable docker` – Esse commando configurou o Docker para iniciar automaticamente toda vez que o Ubuntu for ligado.
- `sudo usermod -aG docker $USER` – Esse comando adicionou o usuário atual ao grupo de permissões chamado `docker`. Esses últimos três commands podem ser visualizados na figura 3.

```
daiane@daiane-VirtualBox: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
Configurando runc (1.2.5-0ubuntu1-24.04.1) ...  
Configurando liberror-perl (0.17029-2) ...  
Configurando bridge-utils (1.7.1-1ubuntu2) ...  
Configurando pigz (2.8-1) ...  
Configurando git-man (1:2.43.0-1ubuntu7.2) ...  
Configurando containerd (1.7.27-0ubuntu1-24.04.1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →  
  /usr/lib/systemd/system/containerd.service.  
Configurando ubuntu-fan (0.12.16) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service →  
  /usr/lib/systemd/system/ubuntu-fan.service.  
Configurando docker.io (27.5.1-0ubuntu3-24.04.2) ...  
info: Selecionando GID da faixa 100 a 999 ...  
info: Adicionando grupo 'docker' (GID 127) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /us  
r/lib/systemd/system/docker.service.  
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/li  
b/systemd/system/docker.socket.  
Configurando git (1:2.43.0-1ubuntu7.2) ...  
A processar 'triggers' para man-db (2.12.0-4build2) ...  
daiane@daiane-VirtualBox:~$ sudo systemctl start docker  
daiane@daiane-VirtualBox:~$ sudo systemctl enable docker  
daiane@daiane-VirtualBox:~$ sudo usermod -aG docker $USER  
daiane@daiane-VirtualBox:~$
```

Figura 3. Iniciando, configurando e adicionando usuário no docker.

- `sudo reboot` – Esse comando foi aplicado para executar as permissões corretamente. Para isso, ele reinicia o sistema.

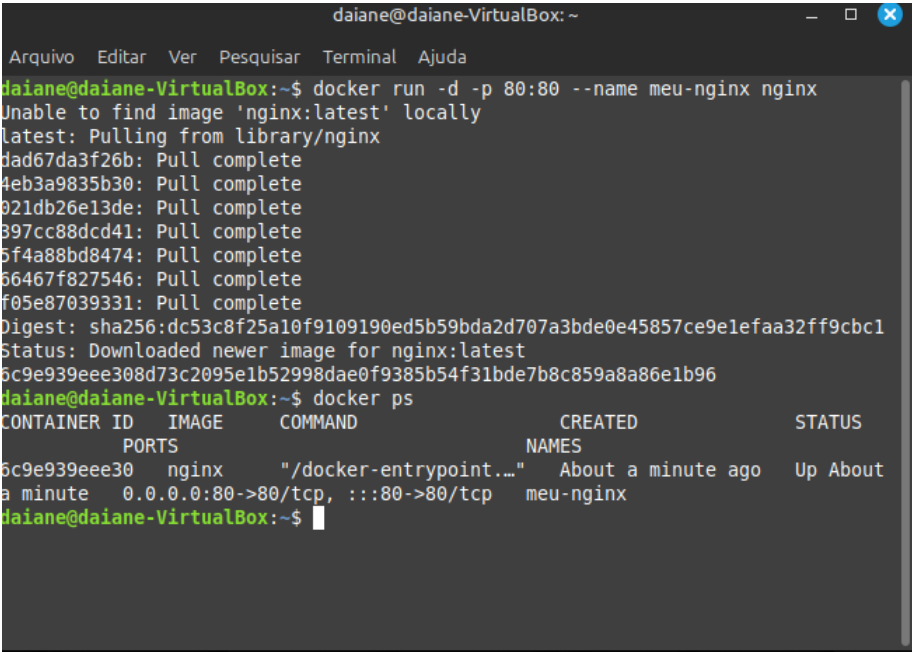
Depois que o Sistema foi reiniciado, o passo a passo descrito nessa etapa 1 foi corretamente implementado. Então, foi possível criar um container com Nginx para ser

usado a partir do docker dentro do VirtualBox. O passo a passo para a criação do container com Nginx está descrito na etapa 2.

2.2. Etapa 2 - Criando um container com o NGINX

1. No terminal do VirtualBox, executamos os comandos:

- `docker run -d -p 80:80 --name meu-nginx nginx` – Esse comando criou e iniciou um container do Docker com o servidor web NGINX rodando na porta 80, como mostra a figura 4. O **nome personalizado** do container criado é: `meu-nginx`.
- `docker ps` – Esse comando mostrou todos os containers do Docker que estavam em execução (ver figura 4).



```
daiane@daiane-VirtualBox: ~  
Arquivo Editar Ver Pesquisar Terminal Ajuda  
daiane@daiane-VirtualBox:~$ docker run -d -p 80:80 --name meu-nginx nginx  
Unable to find image 'nginx:latest' locally  
latest: Pulling from library/nginx  
dad67da3f26b: Pull complete  
4eb3a9835b30: Pull complete  
921db26e13de: Pull complete  
897cc88dcd41: Pull complete  
5f4a88bd8474: Pull complete  
66467f827546: Pull complete  
f05e87039331: Pull complete  
Digest: sha256:dc53c8f25a10f9109190ed5b59bda2d707a3bde0e45857ce9e1efaa32ff9cbc1  
Status: Downloaded newer image for nginx:latest  
6c9e939eee308d73c2095e1b52998dae0f9385b54f31bde7b8c859a8a86e1b96  
daiane@daiane-VirtualBox:~$ docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS  
6c9e939eee30   nginx     "/docker-entrypoint..." About a minute Up About a minute  
0.0.0.0:80->80/tcp, :::80->80/tcp   meu-nginx  
daiane@daiane-VirtualBox:~$
```

Figura 4. Criação do container com o Nginx e verificação dos containers em execução, respectivamente.

Para acessar o IP da MV por meio do navegador de qualquer dispositivo, nós mudamos a configuração de Rede na MV, conforme descrito na etapa 3.

2.3. Etapa 3 – Alterando a configuração de rede no VirtualBox

1. Desligamos a MV no VirtualBox.
2. No VirtualBox, selecionamos a Máquina Virtual Linux2025 e clicamos em Configurações.
3. Em seguida, selecionamos Rede e Adaptador1.
4. Na opção “Conectado a”, escolhemos Placa em modo bridge.
5. Em “Nome”, selecionamos a placa de rede Wi-Fi do notebook.
6. Clicamos em Ok e iniciamos a MV novamente.

Todos esses passos podem ser visualizados na figura 5.

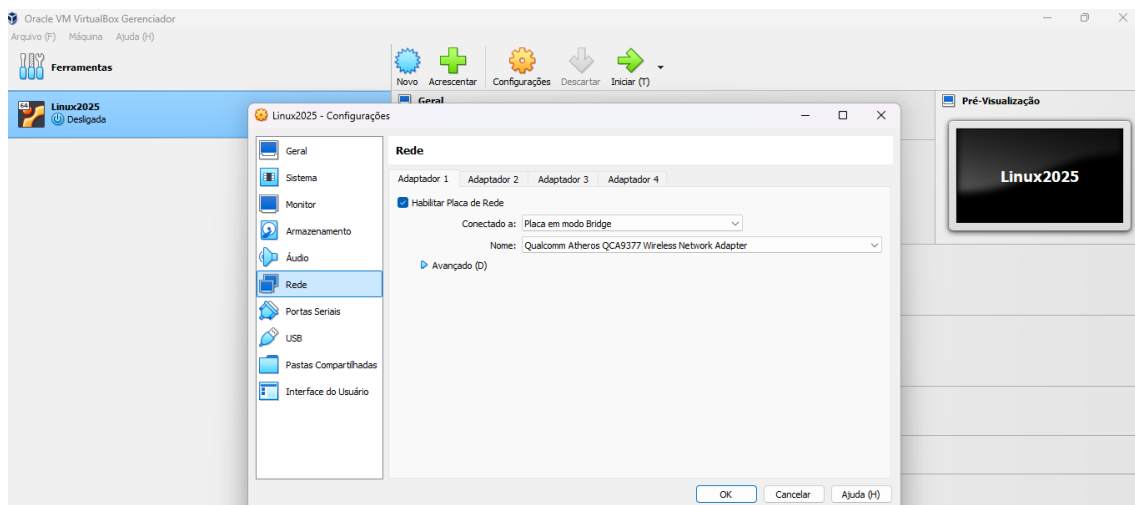


Figura 5. Configuração de rede no VirtualBox.

2.4. Etapa 4 – Descobrimos o IP da MV e acessando o Nginx pelo navegador de qualquer dispositivo.

1. Com a MV inicializada, executamos no terminal o comando: `ip a` - Esse comando nos apresentou o IP da MV. O IP da MV está destacado na figura 6.

```
daiane@daiane-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:12:90:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.4/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 7114sec preferred_lft 7114sec
    inet6 2804:29b8:5071:2896:b8e1:47c9:9d93:1ef2/64 scope global temporary dynamic
        valid_lft 259198sec preferred_lft 86000sec
    inet6 2804:29b8:5071:2896:29da:5769:c03d:bf90/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 259198sec preferred_lft 172798sec
    inet6 fe80::ed63:ee2:9783:e3ff/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:60:1c:26:69 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:60ff:fe1c:2669/64 scope link
        valid_lft forever preferred_lft forever
5: veth917dfd9@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether 76:33:67:e6:87:76 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::7433:67ff:fee6:8776/64 scope link
        valid_lft forever preferred_lft forever
```

Figura 6. Localizando o IP da MV.

A **enp0s3** é a interface de rede NAT padrão do VirtualBox. O **inet**, que está abaixo da interface **enp0s3**, mostra o endereço IPv4 da MV: 192.168.1.4.

2. Em seguida, digitamos `http:// 192.168.1.4`, no navegador do notebook e visualizamos a imagem apresentada na figura 7.

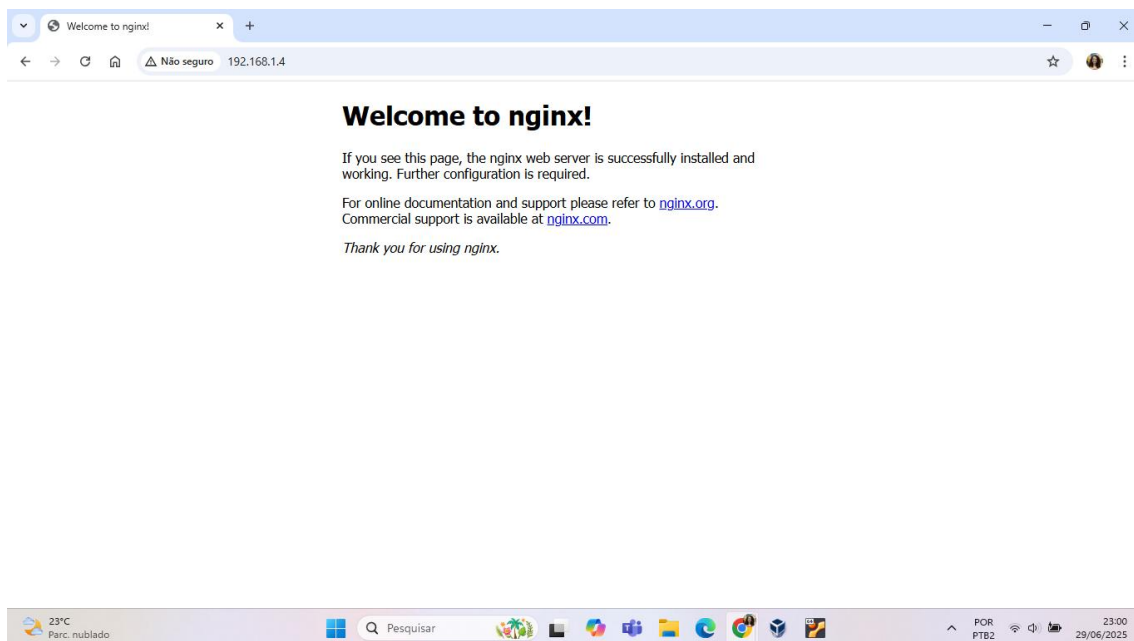


Figura 7. Localizando o IP da MV.

A figura 7 mostra a página padrão do NGINX, indicando que o container com o Nginx foi instalado com sucesso na MV.

Em seguida, para personalizar o código da página `index.html`, tivemos que acessar o container do Nginx. Essa personalização está descrita na página 5.

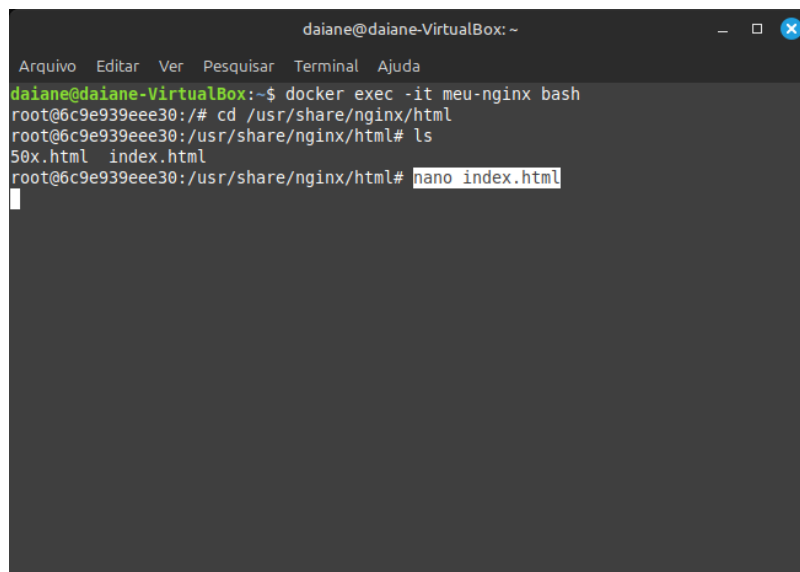
2.5. Etapa 5 – Personalizando a página `index.html` e criando novas pastas no Nginx.

1. No terminal da MV, executamos os comandos:

- `docker exec -it meu-nginx bash` – Esse comando abriu um terminal dentro do container chamado `meu-nginx`.
- `cd /usr/share/nginx/html` – Dentro do container, esse comando localizou a página onde estava o `index.html`.
- `ls` – Esse comando listou os arquivos da pasta.
- `nano index.html` – Com esse comando nós personalizamos o arquivo `index.html` com o editor `nano`.

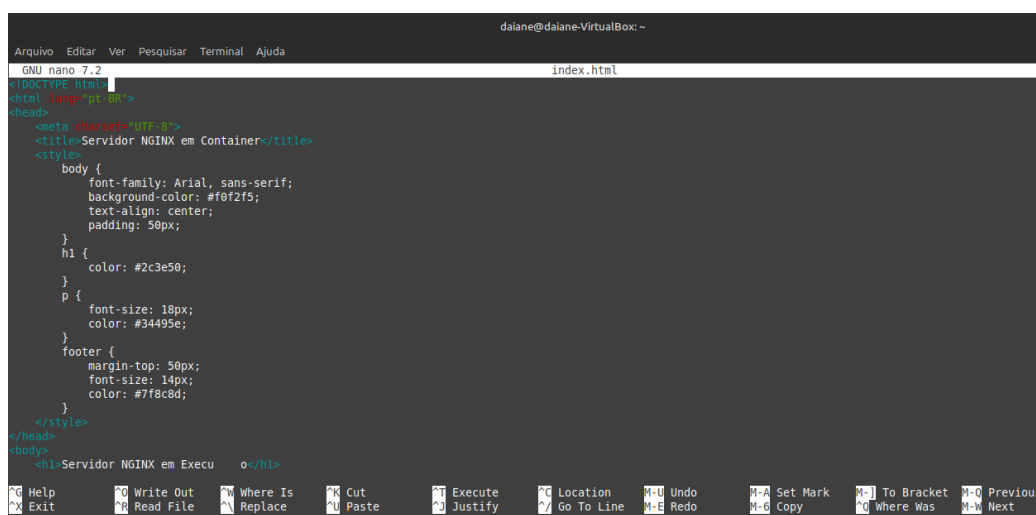
A execução de todos os comandos listados para essa etapa 5, estão ilustrados nas figuras 8 e 9.

Para salvar o arquivo `index.html` pressionamos `Ctrl + O` no teclado e em seguida pressionamos a tecla `Enter`. Para sair do referido arquivo pressionamos `Ctrl + X`.



```
daiane@daiane-VirtualBox: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
daiane@daiane-VirtualBox:~$ docker exec -it meu-nginx bash
root@6c9e939eee30:/# cd /usr/share/nginx/html
root@6c9e939eee30:/usr/share/nginx/html# ls
50x.html index.html
root@6c9e939eee30:/usr/share/nginx/html# nano index.html
```

Figura 8. Acessando o arquivo index.html.



```
GNU nano 7.2 index.html
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta char-set="UTF-8">
  <title>Servidor NGINX em Container</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f2f5;
      text-align: center;
      padding: 50px;
    }
    h1 {
      color: #2c3e50;
    }
    p {
      font-size: 18px;
      color: #34495e;
    }
    footer {
      margin-top: 50px;
      font-size: 14px;
      color: #7f8c8d;
    }
  </style>
</head>
<body>
  <h1>Servidor NGINX em Execu    o</h1>
```

Figura 9. Personalizando no nano o arquivo index.html.

2. Para mostrar que é possível acessar o Nginx através do navegador de qualquer dispositivo, depois da atualização do arquivo index.html, nós acessamos o Nginx pelo navegador de um celular, a página atualizada pode ser verificada pela **figura 10**(Falta colocar a figura 10).

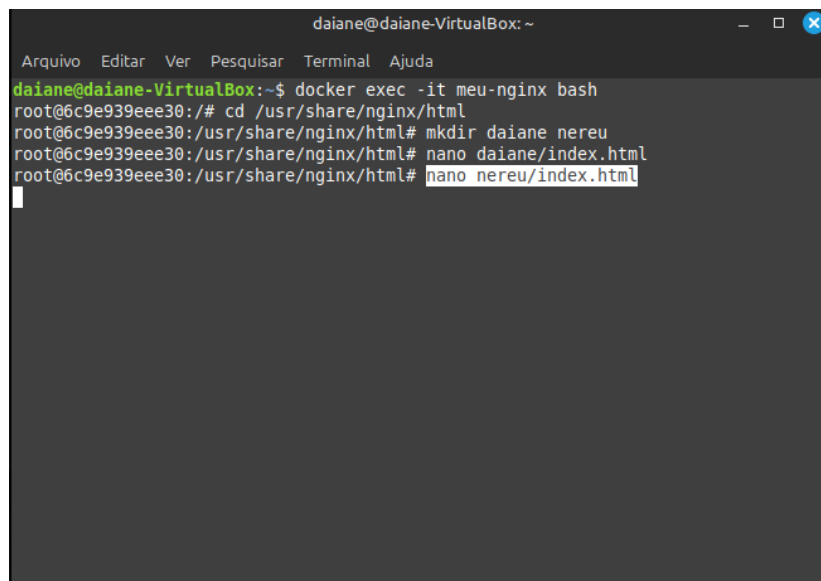
3. Em seguida, criamos as pastas daiane e nereu dentro do container, no mesmo diretório onde está o index.html principal, que é servido pelo NGINX. Em cada uma das pastas (daiane, nereu) foram criados arquivos index.html, que podem ser acessados através do

index.html principal. Para criarmos essas duas pastas, com os arquivos index.html, executamos no terminal da MV os comandos:

- `docker exec -it meu-nginx bash` - Esse comando abriu um terminal dentro do container chamado meu-nginx.
- `cd /usr/share/nginx/html` - Esse comando acessou o diretório HTML do Nginx.
- `mkdir daiane nereu` - Esse comando criou as pastas daiane e nereu.
- `nano daiane/index.html` - Por meio desse comando criamos e personalizamos a pasta e o arquivo daiane/index.html. Em seguida, salvamos o arquivo com `Ctrl + O` e saímos do arquivo com `Ctrl + X`.
- `nano nereu/index.html` - Com esse comando criamos e personalizamos a pasta e o arquivo nereu/index.html. Em seguida, salvamos o arquivo com `Ctrl + O` e saímos do arquivo com `Ctrl + X`.

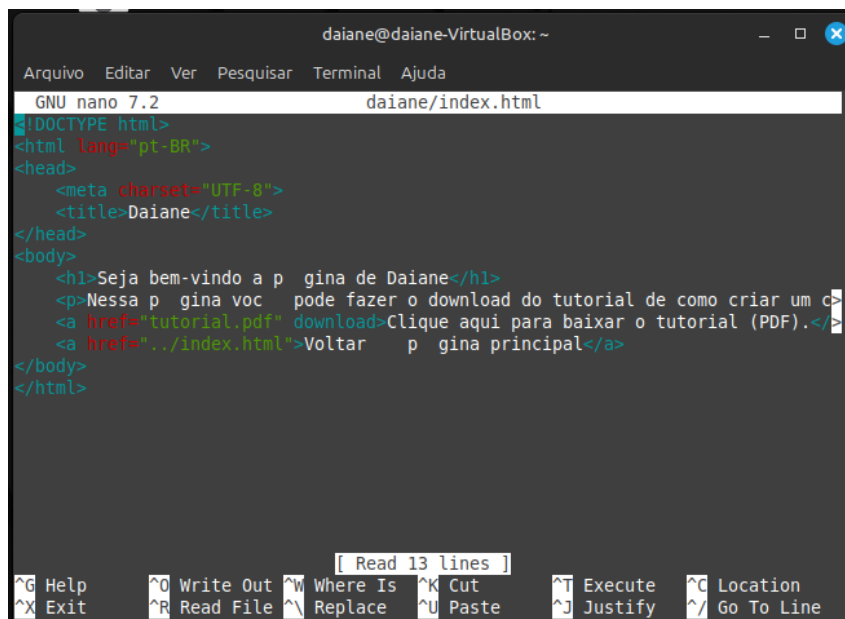
A execução de todos os comandos descritos nesse passo está ilustrada nas figuras 11, 12 e 13.

Depois, atualizamos o index.html principal com os links do index.html da pasta daiane e nereu.



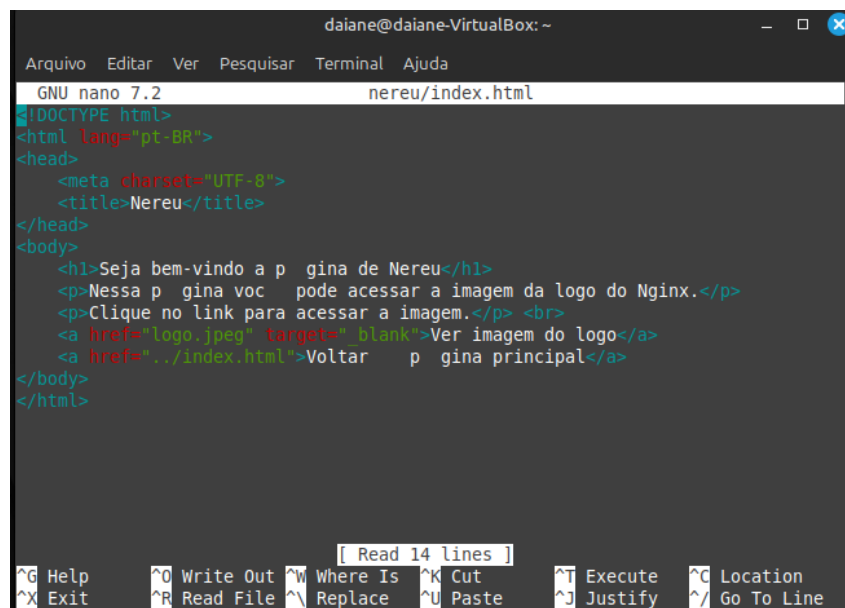
```
daiane@daiane-VirtualBox: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
daiane@daiane-VirtualBox:~$ docker exec -it meu-nginx bash  
root@6c9e939eee30:/# cd /usr/share/nginx/html  
root@6c9e939eee30:/usr/share/nginx/html# mkdir daiane nereu  
root@6c9e939eee30:/usr/share/nginx/html# nano daiane/index.html  
root@6c9e939eee30:/usr/share/nginx/html# nano nereu/index.html
```

Figura 11. Criando a pasta daiane e nereu e o arquivo index em cada pasta.



```
daiane@daiane-VirtualBox: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
GNU nano 7.2 daiane/index.html
!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Daiane</title>
</head>
<body>
  <h1>Seja bem-vindo a p gina de Daiane</h1>
  <p>Nessa p gina voc pode fazer o download do tutorial de como criar um c</p>
  <a href="tutorial.pdf" download>Clique aqui para baixar o tutorial (PDF).</a>
  <a href=" ../index.html">Voltar p gina principal</a>
</body>
</html>
[ Read 13 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Figura 12. Personalizando o index.html na pasta daiane.



```
daiane@daiane-VirtualBox: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
GNU nano 7.2 nereu/index.html
!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Nereu</title>
</head>
<body>
  <h1>Seja bem-vindo a p gina de Nereu</h1>
  <p>Nessa p gina voc pode acessar a imagem da logo do Nginx.</p>
  <p>Clique no link para acessar a imagem.</p> <br>
  <a href="logo.jpeg" target=" blank">Ver imagem do logo</a>
  <a href=" ../index.html">Voltar p gina principal</a>
</body>
</html>
[ Read 14 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Figura 13. Personalizando o index.html na pasta daiane.

Pelas figuras 12 e 13 é possível perceber que nós copiamos arquivos externos para dentro do container com o Nginx. No arquivo index.html da pasta daiane, podemos acessar esse tutorial em pdf e no arquivo index.html da pasta nereu, é possível acessar a imagem da logo do Nginx. Iremos explicar como fizemos isso no passo 4.

4. O arquivo tutorial.pdf e logo. Jpeg foram salvos na MV, na pasta download. Então, executamos no terminal o comando:

- `docker cp /home/daiane/Downloads/tutorial.pdf meu-nginx:/usr/share/nginx/html/daiane/tutorial.pdf` – Esse comando foi usado fora do container para mover o arquivo tutorial.pdf para a

pasta /usr/share/nginx/html/daiane/ dentro do container meu-nginx.

- `docker cp /home/daiane/Downloads/logo.jpeg meu-nginx:/usr/share/nginx/html/nereu/logo.jpeg` – Esse comando foi usado fora do container para mover o arquivo logo.jpeg para a pasta /usr/share/nginx/html/nereu/ dentro do container meu-nginx.

A figura 14 mostra a implementação dos comandos acima, funcionando corretamente.

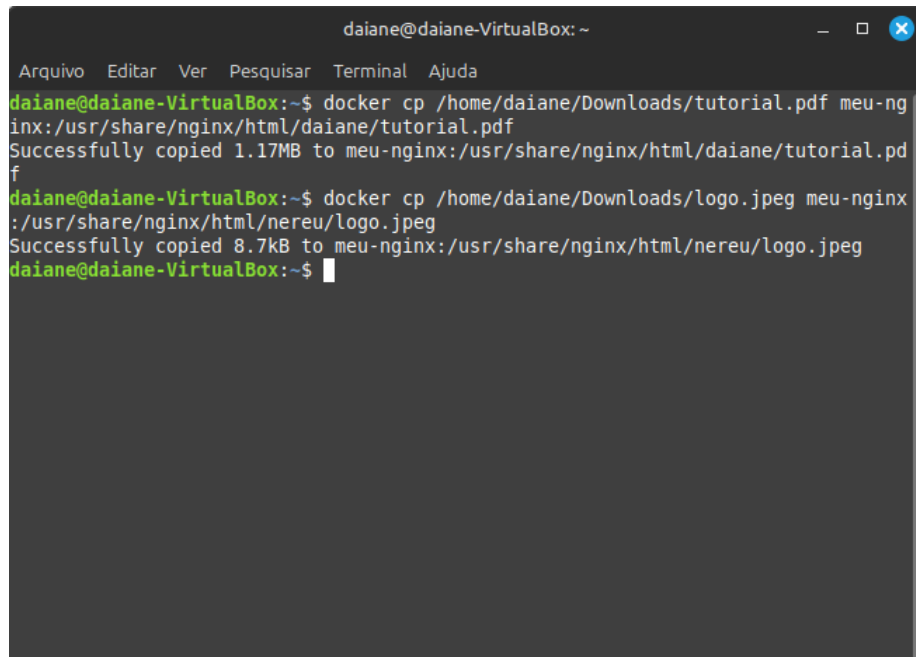
A screenshot of a terminal window titled 'daiane@daiane-VirtualBox: ~'. The window has a menu bar with 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The terminal shows two Docker commands being executed. The first command is 'docker cp /home/daiane/Downloads/tutorial.pdf meu-nginx:/usr/share/nginx/html/daiane/tutorial.pdf', which results in 'Successfully copied 1.17MB to meu-nginx:/usr/share/nginx/html/daiane/tutorial.pdf'. The second command is 'docker cp /home/daiane/Downloads/logo.jpeg meu-nginx:/usr/share/nginx/html/nereu/logo.jpeg', which results in 'Successfully copied 8.7kB to meu-nginx:/usr/share/nginx/html/nereu/logo.jpeg'. The prompt 'daiane@daiane-VirtualBox:~\$' is visible at the bottom.

Figura 14. Copiando o arquivo tutorial.pdf e logo.jpeg para dentro do container.

Assim, foi possível acessar o arquivo tutorial.pdf pela página index.html da pasta daiane e acessar o arquivo logo.jpeg pela página index.html da pasta nereu, **como mostra as figuras 15 e 16**, respectivamente.

Referências

W3C – WORLD WIDE WEB CONSORTIUM. (2023) “HTTP – Hypertext Transfer Protocol Overview”, <https://www.w3.org/Protocols/>, May.