

Tutorial instalar filebrowser com docker compose

Link: <https://viniciuspaes.com/docker/tutorial-instalar-filebrowser-docker-compose/>
<https://github.com/filebrowser/filebrowser>

Objetivo

Instalar imagem do filebrowser em formato de container com o docker. Neste exemplo vamos instalar em um ambiente linux, mas o setup é o mesmo para o windows.

Requisitos

1. Software docker instalado e configurado no sistema operacional
2. Docker compose instalado e configurado
3. Conexão com internet para que seja possível realizar o download da imagem do filebrowser

Como instalar imagem do filebrowser com docker

Passo 1: Ter uma pasta dedicada para gerenciar os containers

Caso esteja configurando em linha de comando no linux:

```
mkdir ~/docker  
cd ~/docker
```

Passo 2: Criar uma pasta para armazenar os arquivo de configuração do filebrowser

```
mkdir ~/docker/filebrowser
```

Passo 3: Criar arquivos de configuração em branco do filebrowser

Na primeira execução/instalação do **filebrowser**, é necessário criar os arquivos de configuração em branco, para evitar problemas do container criar os mesmos como diretório.

Dentro da pasta **~/docker/filebrowser** vamos criar a pasta **config/** para armazenar os arquivos de configuração (que inicialmente vão estar em branco):

```
mkdir ~/docker/filebrowser/config
touch ~/docker/filebrowser/config/filebrowser.db
touch ~/docker/filebrowser/config/filebrowser.json
```

Passo 4: Descobrindo o UID e o GID do seu usuário no linux

No terminal utilize o comando para descobrir o **UID** (user ID):

```
id -u
```

Para descobrir o **GID** (group ID):

```
id -g
```

No meu exemplo, meu **UID** e **GID** tiveram como resultado o número **1000**.

Passo 5: Definindo corretamente o proprietário dos arquivos de configuração do filebrowser

É necessário que os arquivos `filebrowser.db` e `filebrowser.json` tenham como proprietário o **usuário** e o **grupo de usuário** corretos, no meu caso:

```
sudo chown vinicius:vinicius ~/docker/filebrowser/config/filebrowser.*
```

É possível também utilizar o **UID** e o **GID** no lugar dos nomes do **usuário** e **grupo de usuário** acima, ou seja, **1000:1000** (ao invés de **vinicius:vinicius**).

Passo 6: Criar o arquivo de docker compose para o filebrowser

Agora precisamos criar o arquivo **docker-compose.yml** dentro da pasta **~/docker/filebrowser** com o editor de texto conhecido como nano.

```
nano ~/docker/filebrowser/docker-compose.yml
```

Dentro do **docker-compose.yml** é necessário inserir o conteúdo abaixo:

```
---
version: "2"
services:
  filebrowser:
    image: filebrowser/filebrowser
    container_name: filebrowser
    volumes:
      - /home/vinicius
:/srv #Pasta onde deseja que os arquivos sejam armazenados
      - /home/vinicius/docker/filebrowser/config/filebrowser.db:/database.db
```

```
- /home/vinicius/docker/filebrowser/config/filebrowser.json:/filebrowser.json
ports:
  - 80
:80 #Mapeamento porta e
user: 1000:1000 #(Id do usuário) UID:GID (Id
restart: unless-stopped
```

Após inserir os dados acima dentro do arquivo docker-compose.yml pelo nano, lembre-se de salvar o arquivo e sair do editor.

Passo 7: Inicializar o container

Caso a imagem do container não esteja disponível (possivelmente não estará) o docker fará o download automaticamente.

Para iniciar a imagem do container com o docker compose, utilize o comando:

```
cd ~/docker/filebrowser
docker compose up -d
```

Caso esteja utilizando o filebrowser em seu próprio computador, é possível acessá-lo pelo browser, pela url:

```
http://localhost:porta
```

Ou se estiver utilizando o mesmo em algum computador remoto ou servidor, basta acessar pelo ip:porta, exemplo:

```
http://192.168.1.70:80
```

Caso já tenha algum serviço funcionando na porta 80, é necessário editar o arquivo docker-compose.yml e alterar o mapeamento de portas, exemplo:

```
ports:
  - 8077:80
```

Desta forma o serviço do filebrowser ficará disponível na porta **8077**.

Conclusão

Instalar o container do filebrowser é bastante simples. Necesário ter cuidado apenas na primeira vez que inicializar o container, pois é necessário ter os arquivos de configuração criados previamente, senão o container irá criar os mesmos como diretórios e causará erros de execução.

Para demais comandos utilizando o docker em linha de comando, acesse o artigo:

<https://viniciuspaes.com/raspberry-pi/tutorial-como-instalar-docker-raspberry-pi/>

