

Copiando arquivos para dentro do container

Apr 3, 2019

Às vezes volumes não funcionam e a gente precisa copiar coisas para dentro do container. É sério! Você deve estar se perguntando, "*como uma tecnologia que todo mundo usa, não funciona?!?*"

Tá, tá... Eu sei que isso tá parecendo aquelas histórias de "funciona na minha máquina" invertido. Mas vou explicar, no começo do ano eu estava trabalhando com um computador provisório. Infelizmente, eu não tinha poderes de administrador desse computador, o que me impedia de fazer certas coisas, inclusive de dar permissão ao Docker para compartilhar volumes com o sistema de arquivos do Windows.

Ah sim, talvez seja importante mencionar que eu só descobri como copiar arquivos para dentro de container que tá rodando, pois o Windows tem todas essas paradas de permissão.

Então bora lá!

Materiais

Para essa colinha você vai precisar de:

- Um container Docker rodando
- Um arquivo que você quer copiar para dentro do Docker
- Conhecimentos básicos de Docker

Descobrimo o container

Dependendo de como você inicie o seu container você não vai saber o nome dele, então vamos primeiro conferir isso. Como trabalho com ciência e análise de dados, é muito comum me encontrar com um container do Jupyter rodando, é esse que vou usar aqui. Para pegar o nome do container precisamos listar os containers que estão de pé:

```
1 docker ps -a --format "table {{.ID}}\t{{.Image}}\t{{.Names}}"
```

```
view raw (https://gist.github.com/jtemporal/6ba7e2a2ac369738bb8278ad58993161
/raw/33a400faba6959c4045aeb341354e27e7d191f/docker_command.txt)
docker_command.txt (https://gist.github.com/jtemporal/6ba7e2a2ac369738bb8278ad58993161#file-docker_command-txt)
hosted with ❤ by GitHub (https://github.com)
```

Aqui no meu caso, temos um resultado assim:

CONTAINER ID	IMAGE	NAMES
11b8af1aeb43	jupyter/datascience-notebook	relaxed_hypatia

Eu expliquei como montar esse comando que só mostra o ID do container, a imagem sendo usada e o nome do container nessa colinha aqui (<https://jtemporal.com/brincando-com-a-listagem-de-containers-docker/>). Com isso sabemos que o meu container de interesse se chama `relaxed_hypatia`.

Descobrimos para onde mandar os dados

Bem se, como eu, você normalmente mapearia volumes para compartilhar os dados com seu container provavelmente já sabe para onde mandar os dados. No entanto, se você não faz isso ou é uma imagem nova que você está usando pela primeira vez, sua missão é descobrir para onde mandar os arquivos. Geralmente essa informação está na documentação da imagem.

No caso das imagens do Projeto Jupyter, existe uma pasta tradicional para mapear os volumes que essa: `/home/jovyan/work`. Vou mandar os dados pra lá, então quando você tiver rodando aí pro seu container, lembre-se de substituir esse caminho, para o caminho do seu container.

Finalmente copiando os dados

Agora que você já descobriu o lugar para onde quer mandar os dados e o nome do container, chegou a hora de finalmente colocar os dados lá no container. Se você brinca de copiar arquivos pra lá e pra cá pelo terminal, deve ter o costume de usar o comando `cp`. Mas caso não tenha, o `cp` (de *copy*) é o Ctrl+C Ctrl+V do terminal, ele faz uma cópia de um arquivo em um lugar para outro lugar. Por exemplo, vamos supor que tenho a seguinte situação:

```
pasta1/                                pasta2/
└─ arquivo_A.txt
```

E que eu quero copiar o `arquivo_A.txt` para a `pasta2`, tudo isso pelo terminal. Então eu poderia fazer apenas o seguinte:

```
cp pasta1/arquivo_A.txt pasta2
```

E o resultado disso seria:

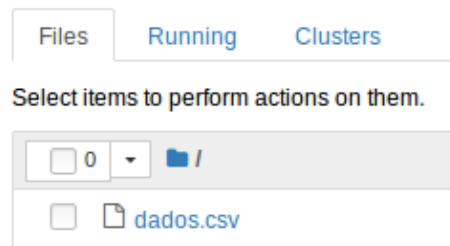
```
pasta1/  
└─ arquivo_A.txt
```

```
pasta2/  
└─ arquivo_A.txt
```

E dá para fazer a mesma coisa com o container. O Docker tem a versão dele chamado `docker cp` que funciona de forma análoga ao `cp` do terminal. Com a pequena diferença que o caminho de destino é formado assim: `nome_do_container:caminho/de/destino`. Então vamos copiar o arquivo `dados.csv` para dentro do meu `relaxed_hypatia`:

```
docker cp dados.csv relaxed_hypatia:/home/jovyan/work/dados.csv
```

E aí eu consigo ver os meus dados lá dentro do meu container. Como no meu caso eu tô rodando um container do Jupyter eu consigo ver esse arquivo lá na minha interface:



Legal né? Você pode seguir a mesma filosofia para retirar dados de dentro do container para sua máquina local, basta inverter a ordem dos caminhos.

Considerações finais







Uma coisa muito importante de lembrar, containers foram feitos para serem efêmeros, então lembre-se de se certificar que está mantendo um backup dos seus dados. Afinal, depois de removido um container não tem mais como recuperar os dados que estavam lá dentro.

Outro motivo que um amigo me ensinou recentemente a favor de copiar os dados para dentro do container é que manter volumes atualizados quando seu container faz muitos processos de leitura e escrita é extremamente custoso. Então, se você tem um arquivo grande que não muda, como um arquivo de dados, ou um processo que faz muita leitura e escrita, como um servidor rails, vale a pena considerar entre copiar os arquivos para dentro do container ou até mesmo fazer uma imagem já com esses arquivos. Lembre-se sempre de pesar os pontos a favor e os pontos contra volumes na próxima vez que for usar containers.

Por hoje é só pessoal. Xêro!

What do you think?

1 Response

 Upvote	 Funny	 Love	 Surprised	 Angry	 Sad
0	0	1	0	0	0

0 Comentários **jtemporal**

1 Entrar ▾

 Recomendar  Tweet  Compartilhar

Ordenar por Mais votados ▾



Iniciar a discussão...

FAZER LOGIN COM



OU REGISTRE-SE NO DISQUS 

Nome

Seja o primeiro a comentar.

TAMBÉM EM JTEMPORAL

Subindo imagens docker pro dockerhub

1 comentário • 5 meses atrás

Avatar **nicero lopes** — Top!!! Só fiquei com uma dúvida...

Retrospectiva 2017: Um momento de reflexão

4 comentários • um ano atrás

Avatar **Fernando Ike** — Escrevi sobre isso também mas não tão bacana. 😊

Retrospectiva 2018: Você cresceu esse ano?



1 comentário • 3 meses atrás

Avatar **Rafael Gomes** — Parabéns pelo ano corrido!

Gerando chaves das APIs do Google

1 comentário • um ano atrás

Avatar **Armando G.** — Eu estou tendo problemas para utilizar essa api do google, não sei porque, eu peguei a chave da api do google, e tentei ...

 Inscreva-se  Adicione o Disqus no seu site [Adicionar Disqus](#) [Adicionar](#)

Brincando com a listagem de containers Docker » (<https://jtemporal.com/brincando-com-a-listagem-de-containers-docker/>)



Recent articles

Brincando com a listagem de containers Docker (<https://jtemporal.com/brincando-com-a-listagem-de-containers-docker/>) 16 Mar 2019

Diferenciando json.loads de json.load e uma pitada de BigQuery (<https://jtemporal.com/diferenciando-json-loads-de-json-load-e-uma-pitade-de-bigquery/>) 15 Mar 2019

Um passeio pelo backstage de um podcast (<https://jtemporal.com/um-passeio-pelo-backstage-de-um-podcast/>) 13 Mar 2019

All rights reserved by jtemporal (<https://jtemporal.com>)

Built by [webjeda](http://webjeda.com) (<http://webjeda.com>)

 (<http://twitter.com/jesstemporal>)  (<http://github.com/jtemporal>)  (<mailto:jessicatemporal@gmail.com>)