



Como manipular Containers

Toda a interação com um **container** vem da linha de comando (*geralmente*), **eles** possuem **ID's** justamente para poderem ser **manipulados de forma individual**, ou seja, se eu quiser parar um **container** posso usar o comando **"docker stop [idContainer]"**, da mesma forma posso reiniciá-lo com **"docker start [idContainer]"**:

```
lucas-ubuntu@lucasubuntu-desktop: ~  
lucas-ubuntu@lucasubuntu-desktop:~$ docker container ls -a  
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS              PORTS          NAMES  
ffb2f61fcc6    ubuntu    "sleep 1d"              32 seconds ago Up 32 seconds          vibrant_babbage  
afec28ffa058    ubuntu    "bash"                  3 hours ago    Exited (0) 3 hours ago    reverent_bhabha  
ff210650013a    hello-world "/hello"                4 hours ago    Exited (0) 4 hours ago    pedantic_pascal  
cedae913513f    hello-world "/hello"                5 hours ago    Exited (0) 5 hours ago    eloquent_moser  
lucas-ubuntu@lucasubuntu-desktop:~$ docker stop ffb2f61fcc6  
ffb2f61fcc6  
lucas-ubuntu@lucasubuntu-desktop:~$ docker ps -a  
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS              PORTS          NAMES  
ffb2f61fcc6    ubuntu    "sleep 1d"              3 minutes ago    Exited (137) 2 minutes ago    vibrant_babbage  
afec28ffa058    ubuntu    "bash"                  3 hours ago    Exited (0) 3 hours ago    reverent_bhabha  
ff210650013a    hello-world "/hello"                4 hours ago    Exited (0) 4 hours ago    pedantic_pascal  
cedae913513f    hello-world "/hello"                5 hours ago    Exited (0) 5 hours ago    eloquent_moser  
lucas-ubuntu@lucasubuntu-desktop:~$ docker start ffb2f61fcc6  
ffb2f61fcc6  
lucas-ubuntu@lucasubuntu-desktop:~$ docker ps -a  
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS              PORTS          NAMES  
ffb2f61fcc6    ubuntu    "sleep 1d"              3 minutes ago    Up 6 seconds          vibrant_babbage  
afec28ffa058    ubuntu    "bash"                  3 hours ago    Exited (0) 3 hours ago    reverent_bhabha
```

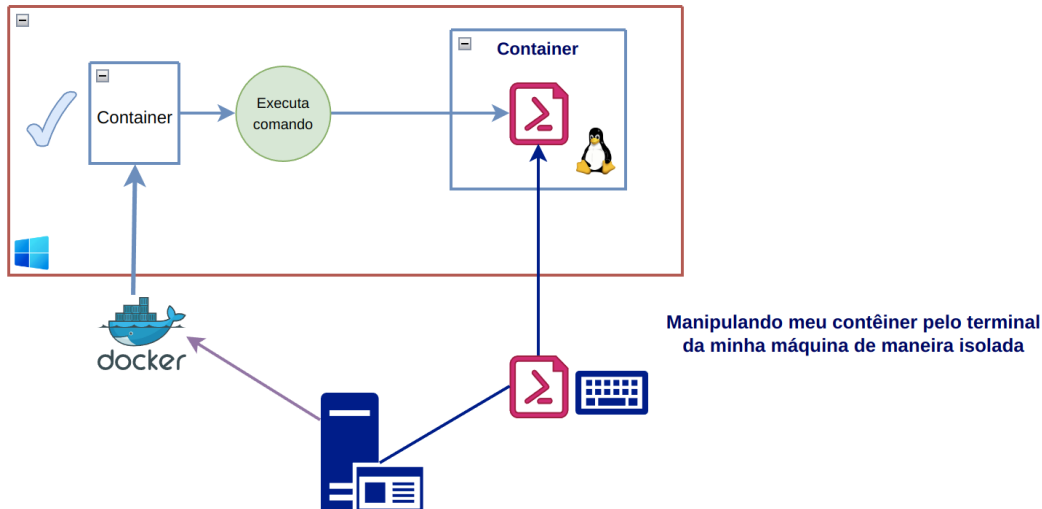


MAS EU QUERO ENTRAR NO MEU CONTAINER!!!

Nem sempre é legal só mandar o **container** executar algo e morrer, as vezes eu preciso **entrar dentro do meu container** e executar as coisas lá dentro (*vibe provisionamento*) e é óbvio que o Docker me permite fazer isso.

Eu posso usar um comando específico de **"docker exec -it [idContainer] bash"**. Esse bash no final é um comando que especifica o que eu quero fazer, nesse caso eu só abri o terminal, a anotação **-it** diz que vou fazer uma **interação (i) no terminal (t)**:

```
lucas-ubuntu@lucasubuntu-desktop:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
ffb2f61fccc6   ubuntu    "sleep 1d"              16 minutes ago Up 13 minutes          vibrant_babbage
afec28ffa058   ubuntu    "bash"                  4 hours ago   Exited (0) 4 hours ago   reverent_bhabha
ff210650013a   hello-world "/hello"                4 hours ago   Exited (0) 4 hours ago   pedantic_pascal
cedae913513f   hello-world "/hello"                6 hours ago   Exited (0) 6 hours ago   eloquent_moser
lucas-ubuntu@lucasubuntu-desktop:~$ docker exec -it [ffb2f61fccc6] bash
root@ffb2f61fccc6:/# cd
root@ffb2f61fccc6:~# touch estou-no-terminal-do-meu-container.txt
root@ffb2f61fccc6:~# ls
estou-no-terminal-do-meu-container.txt
root@ffb2f61fccc6:~#
```



Eu também poderia acessar meu container com um **“docker run -it [nomeContainer] bash”**, só que existe uma **diferença entre “run” e “exec”**. Quando eu uso run eu to iniciando um novo container, quando eu uso “exec” eu estou entrando em um container já existente.

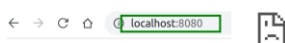


Eu prefiro usar por **Portas!**

Containers podem expor a si mesmos através de portas, existe uma **imagem** no dockerhub interessante para visualizar isso (o único intuito dela é mostrar essa questão de porta no container), se chama **“dockerSample/static-site”**, usando o comando **“docker run -d [nomeImagem]”**, ela aparece **disponível na porta 80**:

```
lucas-ubuntu@lucasubuntu-desktop:~$ docker run -d dockersamples/static-site
Unable to find image 'dockersamples/static-site:latest' locally
latest: Pulling from dockersamples/static-site
fdd5d7827f33: Pull complete
a3ed95caeb02: Pull complete
716f7a5f3082: Pull complete
7b10f03a0309: Pull complete
aff3ab7e9c39: Pull complete
Digest: sha256:daa686c61d7d239b7977e72157997489db49f316b9b9af3909d9f10fd28b2dec
Status: Downloaded newer image for dockersamples/static-site:latest
5ee7463e134302c80f99e6228e3fb4048708815f033f7ab5bc591a620867ba98
lucas-ubuntu@lucasubuntu-desktop:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
5ee7463e1343   dockersamples/static-site "/bin/sh -c 'cd /usr..." 26 seconds ago Up 25 seconds 80/tcp, 443/tcp stupefied_nobel
ffb2f61fccc6   ubuntu    "sleep 1d"              About an hour ago Up About an hour          vibrant_babbage
lucas-ubuntu@lucasubuntu-desktop:~$
```

Mas se eu **acesso a porta 80**, eu recebo um erro:



Não é possível acessar esse site

A conexão com **localhost** foi recusada.

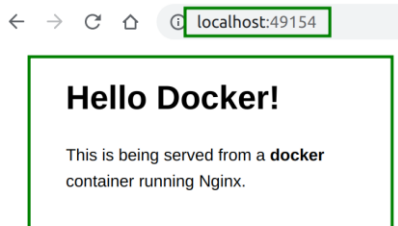
Lembra, **containers são ISOLADOS** da **máquina host**, então é óbvio que eu não vou conseguir acessar a **porta 80 do container** tentando ir pela **porta 80 da MINHA MÁQUINA HOST**. Então o que eu preciso fazer é que meu CONTAINER EXPONHA ESSA PORTA para que eu possa acessá-la.

Da pra fazer isso usando o comando de **run com uma flag -P**, rodando o comando **“docker run -d -P [nomeImagem]”**, já dá pra perceber que a informação da **porta** muda um pouco:

```
lucas-ubuntu@lucasubuntu-desktop:~$ docker run -d -P dockersamples/static-site
29aacbc3698b65869727759afb0d2c28cc03f6be1ffe491ac219c070637586
lucas-ubuntu@lucasubuntu-desktop:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED    STATUS    PORTS
29aacbc3698b   dockersamples/static-site  "/bin/sh -c 'cd /usr_..."  3 seconds ago  Up 2 seconds  0.0.0.0:49154->80/tcp, :::49154->80/tcp, 0.0.0.0:49153->443/tcp, :::49153->443/tcp
f8b2f61fccc6   ubuntu        "sleep id"              2 hours ago  Up 2 hours
```

Se eu usar agora o comando **“docker port [idContainer]”** eu vou conseguir enxergar em **que porta do Container** foi mapeada para ceder acesso em **uma porta da máquina Host**, então se eu digitar **essa porta no meu navegador**, vou conseguir **acessar o conteúdo**:

```
lucas-ubuntu@lucasubuntu-desktop:~$ docker port 29aacbc3698b
443/tcp -> 0.0.0.0:49153
443/tcp -> :::49153
80/tcp -> 0.0.0.0:49154
80/tcp -> :::49154
lucas-ubuntu@lucasubuntu-desktop:~$
```



Da pra fazer isso de maneira ainda mais simples, usando a **“flag -p xxx:xxx”** eu posso dizer que a **porta 8080 do meu container** vai refletir numa **porta X da minha máquina local**, então executando **“docker run -d -p XXXX:XXX [nomeImagem]”** eu vou expor meu container no mapeamento de porta que eu quiser:

