



Certified Tech Developer

The Ultimate Degree

Infraestrutura I

Objetivos

No exercício a seguir, vamos comparar a arquitetura Docker vista anteriormente (PPT Containers e Plataforma Docker) com a operação de um restaurante.

O que temos?

Um Jamboard ilustrando e comparando a operação do Docker com a de um restaurante, só que a ilustração está fora de ordem, também temos o comando para executar nosso primeiro container, que é um servidor web modificado:

[Exercício Infra 1, Aula 4 - Google Jamboard](#)

```
docker container run -d --name spaghetti-docker -p 80:80  
nidio/spaghetti-docker
```

Instruções

Exercício 1

Discutir com a mesa de trabalho quais elementos de ambas as estruturas são semelhantes e reordenar o restaurante e a arquitetura do Docker para que funcionem da mesma forma.

Dica: Imagine qual é a primeira coisa que você faria ao entrar no restaurante. Provavelmente começaremos perguntando pelo cardápio. Qual componente da arquitetura Docker corresponde a esse elemento do restaurante?



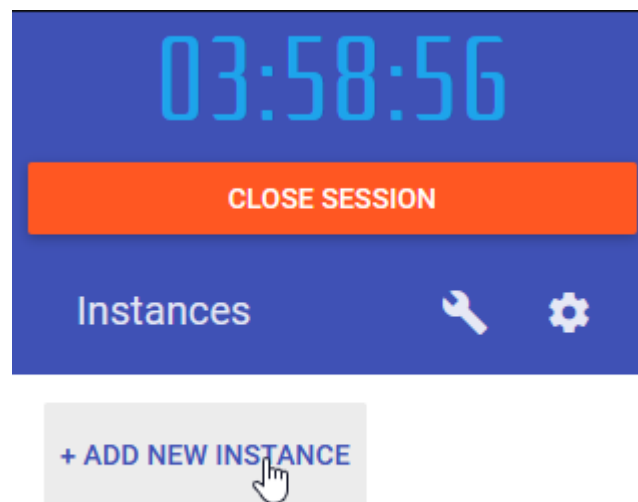
Exercício 2

Agora que entendemos melhor como a arquitetura do Docker funciona, vamos executar nosso primeiro container! Para isso, usaremos a ferramenta Play with Docker. Esta é uma máquina virtual Alpine Linux que executamos em nosso navegador: <https://labs.play-with-docker.com>. Uma vez feito o cadastro no site, clicamos no botão “Iniciar”.





Uma vez dentro, devemos criar uma nova instância:



Aparecerá um terminal na qual executamos nosso primeiro container:

```
#####  
#                               #  
#           WARNING!!!!        #  
# This is a sandbox environment. Using personal credentials #  
# is HIGHLY! discouraged. Any consequences of doing so are #  
# completely the user's responsibilities.                    #  
#                               #  
# The PWD team.                                                #  
#####  
[node2] (local) root@192.168.0.17 ~  
$ docker run -d --name spaghetti-docker -p 80:80 nidio/spaghettdocker  
Unable to find image 'nidio/spaghettdocker:latest' locally  
latest: Pulling from nidio/spaghettdocker  
f7ec5a41d630: Pull complete  
aa1efa14b3bf: Pull complete  
b78b95af9b17: Pull complete  
c7d6bca2b8dc: Pull complete  
cf16cd8e71e0: Pull complete  
0241c68333ef: Pull complete  
4f4fb700ef54: Pull complete  
3bcc5e4b96f9: Pull complete  
e235a205e27d: Pull complete  
Digest: sha256:364f63bde23bcd58460c947e42994160a3f83eb742eb576396400b307f3910ba  
Status: Downloaded newer image for nidio/spaghettdocker:latest  
3c65bd1594b42c8f81d2319fc84a07462ed2df19bdfb25cbac6fe7dd6deabf19  
[node2] (local) root@192.168.0.17 ~  
$
```



Dica: Foi isso que nosso comando fez:

- **docker container run:** Estamos dizendo ao Daemon para executar um container.
- **-d --detach :** executa nosso container em segundo plano.
- **--name:** damos um nome ao container, caso contrário, será atribuído um nome aleatório.
- **-p portaHost:portaContainer:** mapeia as portas do host para a porta do container.
- **nidio/spaghettdocker:** é o endereço onde a imagem base está localizada no Docker Hub (Registry).

Uma vez terminado, vamos abrir a porta do nosso container para ver o servidor web que instalamos:

Na primeira vez, clicamos em **OPEN PORT** e digitamos 80.

labs.play-with-docker.com diz

What port would you like to open?

OK

Cancelar



Depois podemos clicar diretamente no número da porta:

c82k84fn_c82kbjtmrepg008p0krg

IP	192.168.0.17	OPEN PORT	80
Memory	http://ip172-18-0-17-c82k84fnjsv000ch		
	1.23% (49.06MiB / 3.906GiB)		0.78%
SSH			
	ssh ip172-18-0-17-c82k84fnjsv000chaoh0@direct.labs.play-		

DELETEEDITOR

Se você visualizar a tela a seguir, parabéns! Você concluiu o exercício.

Spaghetti Docker

