

Módulo 15 – Docker – Atividade 02

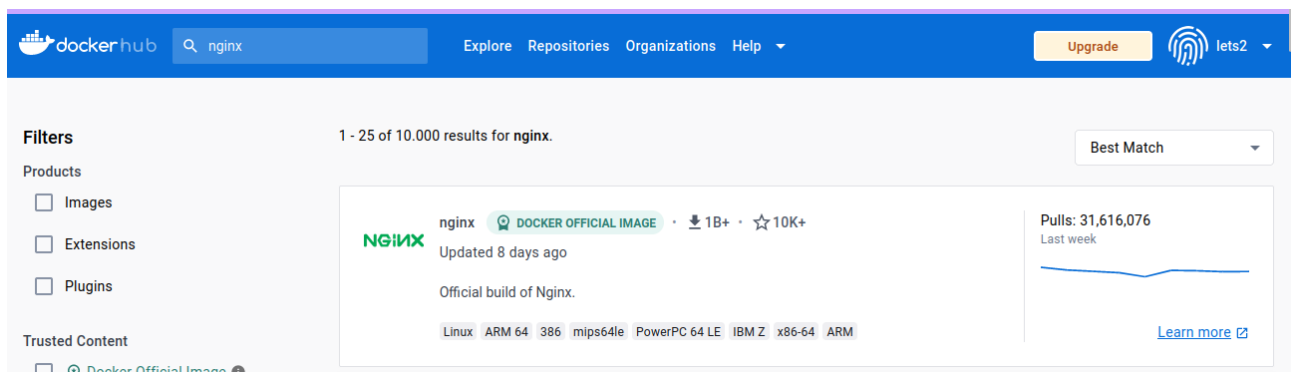
OBJETIVO

Crie um container do nginx e exponha a porta para que seja executada no seu navegador através do link: **http://localhost:8080**. Deve aparecer a página inicial do **Nginx**.

RESOLUÇÃO

Passo 1:

Primeiro, faz-se necessário ter uma conta no Docker Hub. Considerando que já temos uma conta, podemos utilizar a barra de pesquisa para procurar pela imagem do Nginx:



Nota-se diversos resultados, sendo o primeiro uma DOCKER OFFICIAL IMAGE, cujo nome é autoexplicativo. Opcionalmente, a partir de um terminal, podemos utilizar esse comando:

```
docker search nginx
```

A imagem a seguir ilustra o resultado obtido. Nota-se que o primeiro resultado é uma imagem oficial docker:

```
letonio@letonio-Inspiron-15-3567:~$ docker search nginx
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
nginx	Official build of Nginx.	18498	[OK]	
linuxserver/nginx	An Nginx container, brought to you by LinuxS...	196		
bitnami/nginx	Bitnami nginx Docker Image	162		[OK]
ubuntu/nginx	Nginx, a high-performance reverse proxy & we...	89		
privatebin/nginx-fpm-alpine	PrivateBin running on an Nginx, php-fpm & AL...	72		[OK]
bitnami/nginx-ingress-controller	Bitnami Docker Image for NGINX Ingress Contr...	25		[OK]
rancher/nginx-ingress-controller		11		
kasmweb/nginx	An Nginx image based off nginx:alpine and in...	6		
bitnami/nginx-ldap-auth-daemon		3		

Passo 2:

Para fazer o download da imagem nginx, utiliza-se esse comando:

```
docker pull nginx
```

```
letonio@letonio-Inspiron-15-3567:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
9e3ea8720c6d: Downloading [==>] 983kB/31.4MB
bf36b6466679: Downloading [==>] 1.311MB/25.58MB
```

Como a versão desejada não foi informada o docker faz o download da versão mais recente daquela imagem (latest).

Passo 3:

Para verificar as imagens que estão disponíveis na nossa máquina local, utiliza-se esse comando:

`docker image ls`

Na próxima imagem, percebe-se que a imagem nginx faz parte da lista.

```
letonio@letonio-Inspiron-15-3567:~$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
getting-started	latest	b8311f51bdbb	46 hours ago	265MB
<none>	<none>	53e546a1192e	4 days ago	265MB
<none>	<none>	c6f957cba98e	4 days ago	265MB
<none>	<none>	87a860e4e871	4 days ago	265MB
hello-world	latest	9c7a54a9a43c	7 days ago	13.3kB
nginx	latest	448a08f1d2f9	8 days ago	142MB
ex02_docker_and_prisma-prisma-postgres-api	latest	1e2bf16be6a7	11 days ago	363MB

Passo 4:

O próximo passo é criar um container a partir da imagem do nginx que temos disponível. Para tal, utiliza-se o seguinte comando:

`docker run -dp 8080:80 --name first-container-nginx nginx`

```
letonio@letonio-Inspiron-15-3567:~$ docker run -dp 8080:80 --name first-co
ntainer-nginx nginx
e8ae2e1a953ffc0bfff7dbf9853bf29cd36e590d1acaa36e0fa1c7c28aff97837
letonio@letonio-Inspiron-15-3567:~$
```

Sobre o comando acima, run cria o container, a flag -d serve para executar em background (detached), a flag -p serve para mapear a porta 8080 na máquina local (host) com a porta 80 do container. Sem mapear, não teríamos como acessar o container. A flag --name indica o nome de daremos ao container e por fim, nginx indica a imagem que estamos utilizando como base para criar o container.

Passo 5:

Para confirmar que o container está em execução, podemos utilizar esse comando:

`docker ps`

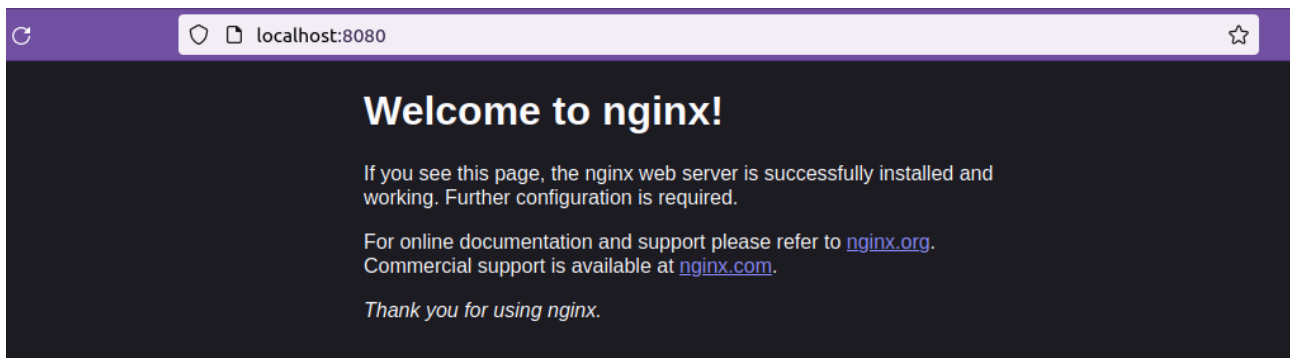
```
letonio@letonio-Inspiron-15-3567:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
e8ae2e1a953f	nginx	"/docker-entrypoint...."	About a minute ago	Up About a minute	0.0.0.0:8080->80/tcp,

```
inner-nginx
letonio@letonio-Inspiron-15-3567:~$
```

Nota-se que o container está em execução a cerca de um minuto. Utilizando o navegador, vamos acessar o endereço:

<http://localhost:8080>



A página inicial do nginx está aparecendo conforme esperado.

Passos opcionais:

Caso seja necessário parar a execução do container, podemos usar o comando:

docker stop (docker stop e8ae2 ou docker stop first-container-nginx).

Para iniciar um container que está parado, podemos usar esse comando:

docker start (docker start e8ae2 ou docker start first-container-nginx).

Por fim, para deletar um container podemos usar o comando:

docker rm (se o container já estiver parado) ou docker rm -f id-do-container.

Problemas encontrados:

Ao tentar parar o container apareceu o seguinte:

```
letonio@letonio-Inspiron-15-3567:~$ docker stop e8a
Error response from daemon: cannot stop container: e8a: permission denied
letonio@letonio-Inspiron-15-3567:~$ sudo docker stop e8a
```

Pelo que foi pesquisado, trata-se de um bug do ubuntu onde o apparmor bloqueia as ações que desligam os dockers criados. Desse modo, recomendou-se utilizar o comando:

sudo aa-remove-unknown

Após executar esse comando, podemos tentar novamente parar o container, utilizando:

docker stop e8a (não precisa o id completo)

```
letonio@letonio-Inspiron-15-3567:~$ sudo docker stop e8a
e8a
```

Mas isso não é tudo, por conta do comando aa-remove-unknown, alguns comandos do terminal podem ser impedidos (por exemplo, code .). Então, precisamos usar os seguintes comandos para voltar ao normal:

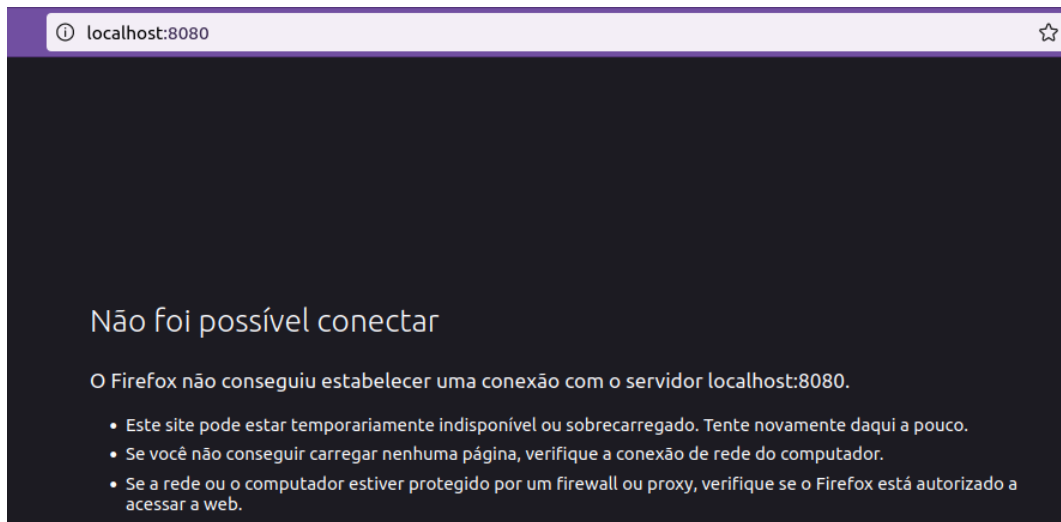
start the apparmor system

sudo systemctl start apparmor

parse and reload all apparmor profiles of installed snap applications

```
sudo apparmor_parser -r /var/lib/snapd/apparmor/profiles/*
```

Desse modo, tudo volta ao normal. Ao parar o container nota-se que:



Para iniciar, utiliza-se:

```
docker start e8a
```

```
letonio@letonio-Inspiron-15-3567:~$ docker stop e8a
e8a
letonio@letonio-Inspiron-15-3567:~$ docker start e8a
e8a
letonio@letonio-Inspiron-15-3567:~$
```

Para remover o container, temos duas alternativas: a) parar o container e, em seguida, usar o comando `docker rm e8a` (ou o nome do container); b) usar a flag `-f` para forçar o docker a parar o container e excluí-lo:

Opção a:

```
docker stop e8a
```

```
docker rm e8a
```

Opção b:

```
docker rm -f e8a (ou docker rm -f first-container-nginx)
```

```
letonio@letonio-Inspiron-15-3567:~$ docker rm -f first-container-nginx
first-container-nginx
letonio@letonio-Inspiron-15-3567:~$
```

Por fim, é importante relembrar que esse comando exclui o container apenas, a imagem continua na máquina local:

```
letonio@letonio-Inspiron-15-3567:~$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
getting-started	latest	b8311f51bdbb	47 hours ago	265MB
<none>	<none>	53e546a1192e	4 days ago	265MB
<none>	<none>	c6f957cba98e	4 days ago	265MB
<none>	<none>	87a860e4e871	4 days ago	265MB
hello-world	latest	9c7a54a9a43c	7 days ago	13.3kB
nginx	latest	448a08f1d2f9	8 days ago	142MB
python3-debian-getting-started	latest	1e2b516ba6a7	11 days ago	265MB