

FIAP GRADUAÇÃO

SISTEMAS DE INFORMAÇÃO

MICROSERVICE AND WEB ENGINEERING

Prof^a. Aparecida Castello Rosa
profaparecida.rosa@fiap.com.br

■ Agenda

- Iniciando com Docker

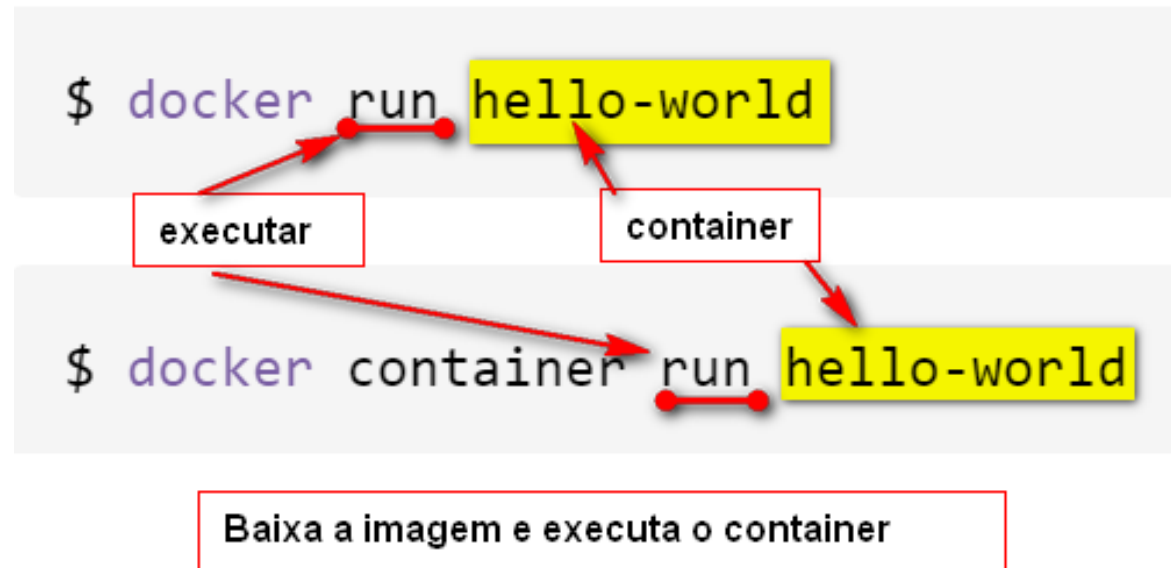
Objetivos

- Iniciando com Docker – parte 2

Docker

Docker

- O Docker possui uma **imagem** personalizada de *hello-world* e serve para testar a instalação e validar se tudo funciona conforme o esperado.



Docker Desktop | Update to latest | Search for images, containers, volumes, extensions and more... | Ctrl+K | Sign in

Images | Give feedback

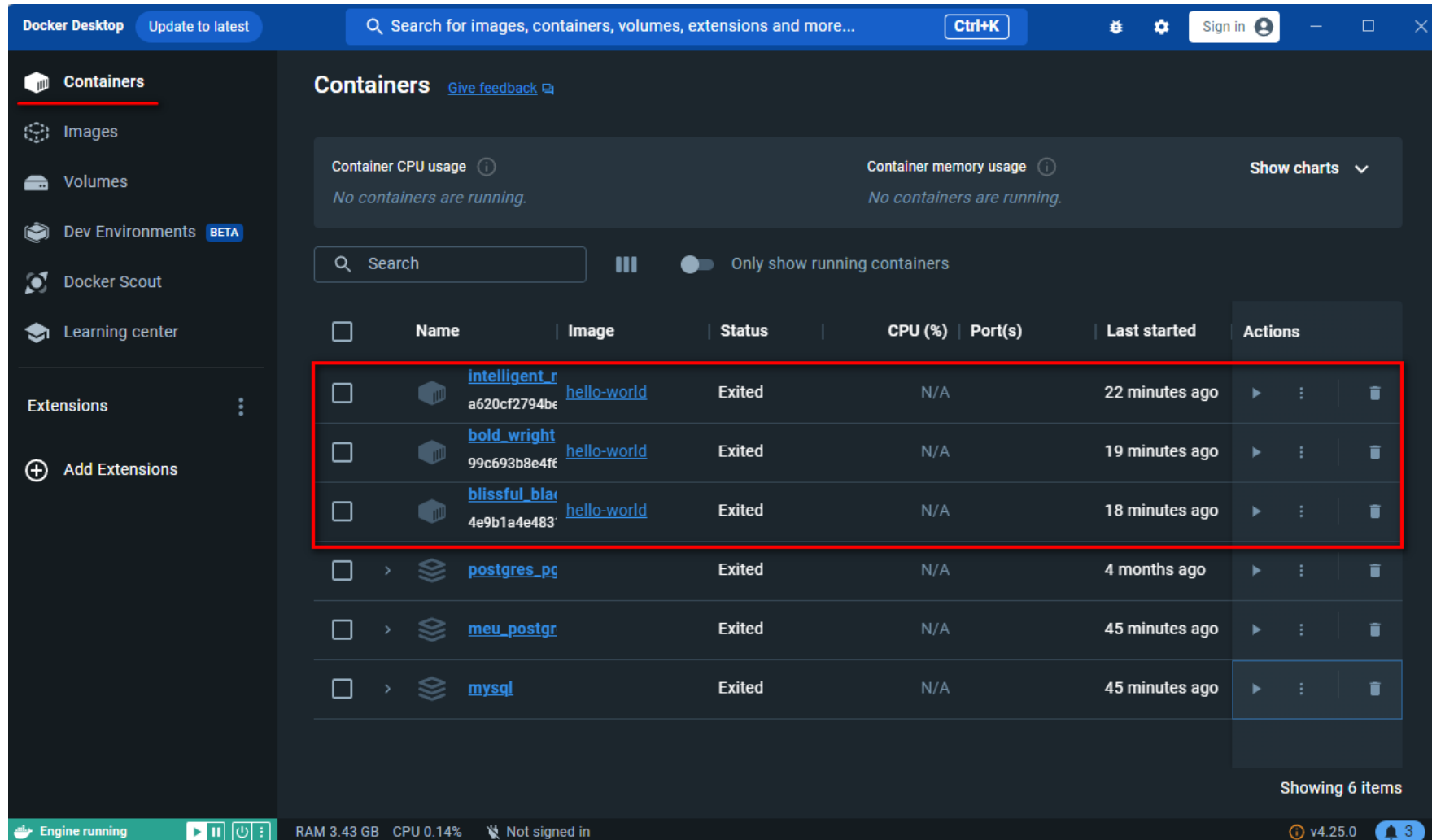
Local | Hub | Artifactory | **EARLY ACCESS**

1.78 GB / 2.15 GB in use | 7 images | Last refresh: 8 hours ago

Search

Name	Tag	Status	Created	Size	Actions
acrdev/pedidos-ms 9561683f2636	latest	Unused	5 months ago	370.56 MB	▶ ⋮ 🗑️
postgres a20f35f462a4	latest	In use	5 months ago	424.88 MB	▶ ⋮ 🗑️
dpage/pgadmin4 da73a5b9ac16	latest	In use	6 months ago	535.15 MB	▶ ⋮ 🗑️
mysql 96bc8cf3633b	8.0	In use	6 months ago	581.8 MB	▶ ⋮ 🗑️
hello-world d2c94e258dcb	latest	In use	1 year ago	13.25 KB	▶ ⋮ 🗑️
openjdk 264c9bdce361	17-alpine	Unused	3 years ago	325.59 MB	▶ ⋮ 🗑️

Showing 7 items



Docker - Listar Imagens

```
Windows PowerShell
PS C:\Users\cida> docker images
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
postgres            14-alpine   8258e2afe6e4  4 months ago 239MB
acrdev/pedidos-ms   latest      9561683f2636  5 months ago 371MB
postgres            latest      a20f35f462a4  5 months ago 425MB
dpage/pgadmin4      latest      da73a5b9ac16  5 months ago 535MB
mysql               8.0         96bc8cf3633b  6 months ago 582MB
hello-world         latest      d2c94e258dcb  12 months ago 13.3kB
openjdk             17-alpine   264c9bdce361  2 years ago  326MB
PS C:\Users\cida> |
```

```
$ docker images
```

```
# ou
```

```
$ docker image ls
```

Label	Descrição
REPOSITORY	O nome da imagem.
TAG	A versão da imagem.
IMAGE ID	Identificação da imagem.
CREATED	Quando a imagem foi criada.
SIZE	Tamanho da imagem.

Docker - Listar Containers

```
Windows PowerShell
PS C:\Users\cida> docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\cida>
```

```
$ docker container ls
#ou
$ docker ps
```

Com o `docker container ls` ou `docker ps`, podemos visualizar todos os containers em execução e ainda obter os detalhes sobre eles.

A saída do `docker container ls` é dividida em sete colunas.

Label	Descrição
CONTAINER ID	Identificação única do <i>container</i> .
IMAGE	A imagem que foi utilizada para a execução do <i>container</i> .
COMMAND	O comando em execução.
CREATED	Quando o <i>container</i> foi criado.
STATUS	O status atual do <i>container</i> .
PORTS	A porta do <i>container</i> e do <i>host</i> que esse <i>container</i> utiliza.
NAMES	O nome do <i>container</i> .

Docker - Listar Containers

```
$ docker container ls
#ou
$ docker ps
```

```
PS C:\Users\cida>
PS C:\Users\cida> docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
1e0a66606281   postgres      "docker-entrypoint.s..." 5 months ago   Up 2 minutes   0.0.0.0:5432->5432/tcp          postgres_container
d329263c1b43   dpage/pgadmin4 "/entrypoint.sh"         5 months ago   Up 2 minutes   443/tcp, 0.0.0.0:5050->80/tcp    pgadmin4_container
093bceaf70fa   mysql:8.0     "docker-entrypoint.s..." 5 months ago   Up 2 minutes   0.0.0.0:3306->3306/tcp, 33060/tcp mysql_container
```

Exibindo todos os **containers** que estão em execução e os detalhes.

Docker - Listar Containers

```
Windows PowerShell
PS C:\Users\cida> docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4e9b1a4e4831	hello-world	"/hello"	34 minutes ago	Exited (0) 28 minutes ago		blissful_blackburn
99c693b8e4f6	hello-world	"/hello"	35 minutes ago	Exited (0) 29 minutes ago		bold_wright
a620cf2794be	hello-world	"/hello"	38 minutes ago	Exited (0) 32 minutes ago		intelligent_margulis
e89ac921429d	dpape/pgadmin4	"/entrypoint.sh"	3 months ago	Exited (0) 3 months ago		dev-pgadmin
aef997f5231c	postgres:14-alpine	"docker-entrypoint.s..."	3 months ago	Exited (0) 3 months ago		dev-postgresql
1e0a66606281	postgres	"docker-entrypoint.s..."	5 months ago	Exited (0) 13 minutes ago		postgres_container
d329263c1b43	dpape/pgadmin4	"/entrypoint.sh"	5 months ago	Exited (0) 13 minutes ago		pgadmin4_container
093bceaf70fa	mysql:8.0	"docker-entrypoint.s..."	5 months ago	Exited (0) 11 minutes ago		mysql_container

```
PS C:\Users\cida> |
```

```
$ docker container ls -a
# ou
$ docker ps -a
```

Com o `docker container ls -a` ou `docker ps -a`, podemos visualizar **todos os containers**, os que estão em execução e os que não estão sendo executados, e os detalhes.

Docker - Listar Containers

```
PS C:\Users\cida>
PS C:\Users\cida> docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a5f21a8097aa	ubuntu	"/bin/bash"	26 hours ago	Exited (0) 26 hours ago		hopeful_turing
e1b261e31fa3	ubuntu	"/bin/bash"	26 hours ago	Exited (0) 26 hours ago		boring_liskov
e89ac921429d	dpage/pgadmin4	"/entrypoint.sh"	3 months ago	Exited (0) 3 months ago		dev-pgadmin
aef997f5231c	postgres:14-alpine	"docker-entrypoint.s..."	3 months ago	Exited (0) 3 months ago		dev-postgresql
1e0a66606281	postgres	"docker-entrypoint.s..."	5 months ago	Up 14 minutes	0.0.0.0:5432->5432/tcp	postgres_container
d329263c1b43	dpage/pgadmin4	"/entrypoint.sh"	5 months ago	Up 14 minutes	443/tcp, 0.0.0.0:5050->80/tcp	pgadmin4_container
093bceaf70fa	mysql:8.0	"docker-entrypoint.s..."	5 months ago	Up 14 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	mysql_container

```
PS C:\Users\cida>
```

```
PS C:\Users\cida>
PS C:\Users\cida> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a5f21a8097aa	ubuntu	"/bin/bash"	26 hours ago	Exited (0) 26 hours ago		hopeful_turing
e1b261e31fa3	ubuntu	"/bin/bash"	26 hours ago	Exited (0) 26 hours ago		boring_liskov
e89ac921429d	dpage/pgadmin4	"/entrypoint.sh"	3 months ago	Exited (0) 3 months ago		dev-pgadmin
aef997f5231c	postgres:14-alpine	"docker-entrypoint.s..."	3 months ago	Exited (0) 3 months ago		dev-postgresql
1e0a66606281	postgres	"docker-entrypoint.s..."	5 months ago	Up 15 minutes	0.0.0.0:5432->5432/tcp	postgres_container
d329263c1b43	dpage/pgadmin4	"/entrypoint.sh"	5 months ago	Up 15 minutes	443/tcp, 0.0.0.0:5050->80/tcp	pgadmin4_container
093bceaf70fa	mysql:8.0	"docker-entrypoint.s..."	5 months ago	Up 15 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	mysql_container

```
PS C:\Users\cida>
```

\$ docker container ls OU \$ docker ps - exibe **containers** em execução.

\$ docker container ls -a OU \$ docker ps -a - exibe todos os **containers**.

Docker - Remover Containers

```
docker rm <id ou name>
```

Se o *container* estiver em execução podemos adicionar a *flag -f* (force) que para o *container* e em seguida o remove da memória local.

```
docker rm <id ou name> --force
```

```
PS C:\Users\cida> docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
4e9b1a4e4831   hello-world    "/hello"                22 hours ago  Exited (0) 22 hours ago          blissful_blackburn
99c693b8e4f6   hello-world    "/hello"                22 hours ago  Exited (0) 22 hours ago          bold_wright
a620cf2794be   hello-world    "/hello"                22 hours ago  Exited (0) 22 hours ago          intelligent_margulis
e89ac921429d   dpag/pgadmin4  "/entrypoint.sh"        3 months ago  Exited (0) 3 months ago          dev-pgadmin
aef997f5231c   postgres:14-alpine "docker-entrypoint.s..." 3 months ago  Exited (0) 3 months ago          dev-postgresql
1e0a66606281   postgres      "docker-entrypoint.s..." 5 months ago  Exited (0) 6 minutes ago          postgres_container
d329263c1b43   dpag/pgadmin4  "/entrypoint.sh"        5 months ago  Exited (0) 6 minutes ago          pgadmin4_container
093bceaf70fa   mysql:8.0      "docker-entrypoint.s..." 5 months ago  Exited (0) 6 minutes ago          mysql_container

PS C:\Users\cida>
PS C:\Users\cida> docker rm 4e9b1a4e4831
4e9b1a4e4831
```

```
b2 C:/Users/cida> docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
0a33c6e94d04   mysql:8.0      "docker-entrypoint.s..." 2 months ago  Exited (0) 24 seconds ago          mysql_container
q35d5e3c1p43   postgres      "docker-entrypoint.s..." 2 months ago  Exited (0) 20 seconds ago          postgres_container
j609e0e0e381   postgres      "docker-entrypoint.s..." 2 months ago  Exited (0) 28 seconds ago          postgres_container
9e4d0d42531c   postgres      "docker-entrypoint.s..." 3 months ago  Exited (0) 3 months ago          dev-postgresql
e89ac921429d   dpag/pgadmin4  "/entrypoint.sh"        3 months ago  Exited (0) 3 months ago          dev-pgadmin
aef997f5231c   postgres      "docker-entrypoint.s..." 5 months ago  Exited (0) 6 minutes ago          postgres_container
d329263c1b43   dpag/pgadmin4  "/entrypoint.sh"        5 months ago  Exited (0) 6 minutes ago          pgadmin4_container
093bceaf70fa   mysql:8.0      "docker-entrypoint.s..." 5 months ago  Exited (0) 6 minutes ago          mysql_container

PS C:/Users/cida> docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
0a33c6e94d04   mysql:8.0      "docker-entrypoint.s..." 2 months ago  Exited (0) 24 seconds ago          mysql_container
q35d5e3c1p43   postgres      "docker-entrypoint.s..." 2 months ago  Exited (0) 20 seconds ago          postgres_container
j609e0e0e381   postgres      "docker-entrypoint.s..." 2 months ago  Exited (0) 28 seconds ago          postgres_container
9e4d0d42531c   postgres      "docker-entrypoint.s..." 3 months ago  Exited (0) 3 months ago          dev-postgresql
e89ac921429d   dpag/pgadmin4  "/entrypoint.sh"        3 months ago  Exited (0) 3 months ago          dev-pgadmin
aef997f5231c   postgres      "docker-entrypoint.s..." 5 months ago  Exited (0) 6 minutes ago          postgres_container
d329263c1b43   dpag/pgadmin4  "/entrypoint.sh"        5 months ago  Exited (0) 6 minutes ago          pgadmin4_container
093bceaf70fa   mysql:8.0      "docker-entrypoint.s..." 5 months ago  Exited (0) 6 minutes ago          mysql_container
```

Docker - Remover Containers

```
PS C:\Users\cida> docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
99c693b8e4f6	hello-world	"/hello"	22 hours ago	Exited (0) 22 hours ago		<u>bold_wright</u>
a620cf2794be	hello-world	"/hello"	22 hours ago	Exited (0) 22 hours ago		intelligent_margulis
e89ac921429d	dpage/pgadmin4	"/entrypoint.sh"	3 months ago	Exited (0) 3 months ago		dev-pgadmin
aef997f5231c	postgres:14-alpine	"docker-entrypoint.s..."	3 months ago	Exited (0) 3 months ago		dev-postgresql
1e0a66606281	postgres	"docker-entrypoint.s..."	5 months ago	Exited (0) 2 minutes ago		postgres_container
d329263c1b43	dpage/pgadmin4	"/entrypoint.sh"	5 months ago	Exited (0) 2 minutes ago		pgadmin4_container
093bceaf70fa	mysql:8.0	"docker-entrypoint.s..."	5 months ago	Exited (0) 2 minutes ago		mysql_container

```
PS C:\Users\cida> docker rm bold_wright
bold_wright
PS C:\Users\cida> docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a620cf2794be	hello-world	"/hello"	22 hours ago	Exited (0) 22 hours ago		intelligent_margulis
e89ac921429d	dpage/pgadmin4	"/entrypoint.sh"	3 months ago	Exited (0) 3 months ago		dev-pgadmin
aef997f5231c	postgres:14-alpine	"docker-entrypoint.s..."	3 months ago	Exited (0) 3 months ago		dev-postgresql
1e0a66606281	postgres	"docker-entrypoint.s..."	5 months ago	Exited (0) 3 minutes ago		postgres_container
d329263c1b43	dpage/pgadmin4	"/entrypoint.sh"	5 months ago	Exited (0) 3 minutes ago		pgadmin4_container
093bceaf70fa	mysql:8.0	"docker-entrypoint.s..."	5 months ago	Exited (0) 2 minutes ago		mysql_container

```
PS C:\Users\cida> |
```


Docker - Remover Containers

```
PS C:\Users\cida> docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a620cf2794be	hello-world	"/hello"	22 hours ago	Exited (0) 22 hours ago		intelligent_margulis
e89ac921429d	dpage/pgadmin4	"/entrypoint.sh"	3 months ago	Exited (0) 3 months ago		dev-pgadmin
aef997f5231c	postgres:14-alpine	"docker-entrypoint.s..."	3 months ago	Exited (0) 3 months ago		dev-postgresql
1e0a66606281	postgres	"docker-entrypoint.s..."	5 months ago	Exited (0) 4 minutes ago		postgres_container
d329263c1b43	dpage/pgadmin4	"/entrypoint.sh"	5 months ago	Exited (0) 4 minutes ago		pgadmin4_container
093bceaf70fa	mysql:8.0	"docker-entrypoint.s..."	5 months ago	Exited (0) 3 minutes ago		mysql_container

```
PS C:\Users\cida> docker rm a62
a62
```

```
PS C:\Users\cida> docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e89ac921429d	dpage/pgadmin4	"/entrypoint.sh"	3 months ago	Exited (0) 3 months ago		dev-pgadmin
aef997f5231c	postgres:14-alpine	"docker-entrypoint.s..."	3 months ago	Exited (0) 3 months ago		dev-postgresql
1e0a66606281	postgres	"docker-entrypoint.s..."	5 months ago	Exited (0) 4 minutes ago		postgres_container
d329263c1b43	dpage/pgadmin4	"/entrypoint.sh"	5 months ago	Exited (0) 4 minutes ago		pgadmin4_container
093bceaf70fa	mysql:8.0	"docker-entrypoint.s..."	5 months ago	Exited (0) 4 minutes ago		mysql_container

```
PS C:\Users\cida> |
```


Docker - Remover *Image*

```
$ docker image rm <Nome ou ID da imagem>  
# ou  
$ docker rmi <id>
```

```
PS C:\Users\cida> docker images  
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE  
postgres             14-alpine   8258e2afe6e4  4 months ago 239MB  
acrdev/pedidos-ms    latest      9561683f2636  5 months ago 371MB  
postgres             latest      a20f35f462a4  5 months ago 425MB  
dpage/pgadmin4       latest      da73a5b9ac16  5 months ago 535MB  
mysql                8.0         96bc8cf3633b  6 months ago 582MB  
hello-world          latest      d2c94e258dcb  12 months ago 13.3kB  
openjdk              17-alpine   264c9bdce361  2 years ago  326MB  
PS C:\Users\cida> docker rmi d2c  
Untagged: hello-world:latest  
Untagged: hello-world@sha256:a26bff933ddc26d5cdf7faa98b4ae1e3ec20c4985e6f87ac0973052224d24302  
Deleted: sha256:d2c94e258dcb3c5ac2798d32e1249e42ef01cba4841c2234249495f87264ac5a  
Deleted: sha256:ac28800ec8bb38d5c35b49d45a6ac4777544941199075dff8c4eb63e093aa81e  
PS C:\Users\cida> docker images  
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE  
postgres             14-alpine   8258e2afe6e4  4 months ago 239MB  
acrdev/pedidos-ms    latest      9561683f2636  5 months ago 371MB  
postgres             latest      a20f35f462a4  5 months ago 425MB  
dpage/pgadmin4       latest      da73a5b9ac16  5 months ago 535MB  
mysql                8.0         96bc8cf3633b  6 months ago 582MB  
openjdk              17-alpine   264c9bdce361  2 years ago  326MB  
PS C:\Users\cida> |
```

Depois que vimos como criar um simples **container**, e como visualizar as **imagens** e **containers** que estão em nosso **host**, vamos criar um novo **container** para usar interativamente.

Parâmetros que são utilizados: `-t`, `-i` e `-d`.

`-t` -- Disponibiliza um TTY (console) para o nosso **container**.

`-i` -- Mantém o STDIN aberto mesmo que você não esteja conectado no **container**.

`-d` -- Faz com que o **container** rode como um *daemon* (ou *detached* - em *background*), ou seja, **sem a interatividade** que os outros dois parâmetros fornecem, liberando o terminal.

Dessa forma temos dois modos de execução de nossos **containers**: modo **interativo** ou modo **detached**.

Docker – Modo interativo

- Listar as imagens e baixar o Ubuntu se necessário.

```
PS C:\Users\cida> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
postgres	14-alpine	8258e2afe6e4	4 months ago	239MB
acrdev/pedidos-ms	latest	9561683f2636	5 months ago	371MB
postgres	latest	a20f35f462a4	5 months ago	425MB
dpage/pgadmin4	latest	da73a5b9ac16	5 months ago	535MB
mysql	8.0	96bc8cf3633b	6 months ago	582MB
openjdk	17-alpine	264c9bdce361	2 years ago	326MB

```
PS C:\Users\cida>
```

Docker – Modo interativo

```
Windows PowerShell
PS C:\Users\cida> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
49b384cc7b4a: Already exists
Digest: sha256:3f85b7caad41a95462cf5b787d8a04604c8262cdcdf9a472b8c52ef83375fe15
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
PS C:\Users\cida> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest             bf3dc08bfed0       8 days ago        76.2MB
postgres            14-alpine          8258e2afe6e4       4 months ago      239MB
acrdev/pedidos-ms   latest             9561683f2636       5 months ago      371MB
postgres            latest             a20f35f462a4       5 months ago      425MB
dpage/pgadmin4      latest             da73a5b9ac16       5 months ago      535MB
mysql               8.0                96bc8cf3633b       6 months ago      582MB
openjdk             17-alpine          264c9bdce361       2 years ago        326MB
PS C:\Users\cida> |
```

I Docker – Modo interativo

Executando o container no modo interativo.

```
$ docker run -it ubuntu bash
```

Dessa forma estamos no *bash* do Ubuntu do **container**.

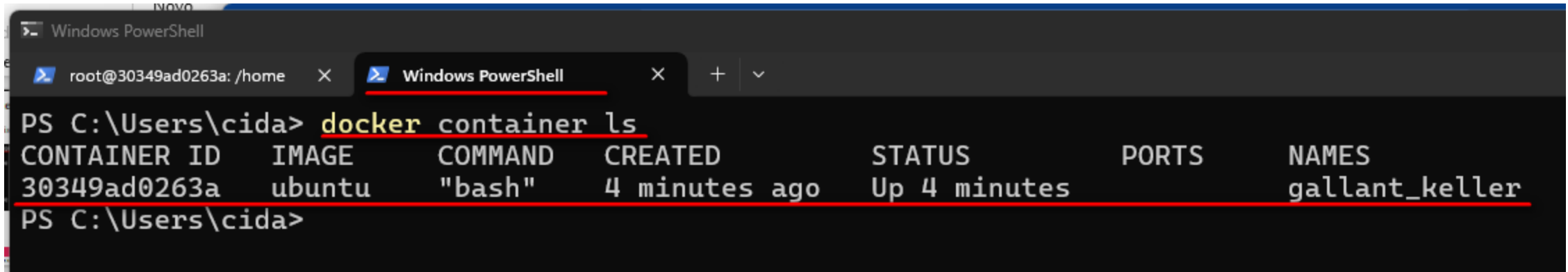
Para sair, digitar o comando `exit`.

```
PS C:\Users\cida> docker run -it ubuntu bash
root@30349ad0263a:/# ls
bin    dev    home   lib64  mnt    proc   run    srv    tmp    var
boot   etc    lib     media  opt     root   sbin   sys    usr
root@30349ad0263a:/# cd home
root@30349ad0263a:/home# ls
ubuntu
root@30349ad0263a:/home# |
```

Docker – Modo interativo

Abrir outro terminal e listar os *containers*.

```
$ docker container ls
```



The screenshot shows a Windows PowerShell terminal window with two tabs. The active tab is titled 'Windows PowerShell' and shows the command 'docker container ls' being executed. The output is a table with columns: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, PORTS, and NAMES. The output shows a single container with ID '30349ad0263a', image 'ubuntu', command '"bash"', created '4 minutes ago', status 'Up 4 minutes', and name 'gallant_keller'.

```
PS C:\Users\cida> docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
30349ad0263a   ubuntu   "bash"    4 minutes ago   Up 4 minutes           gallant_keller
PS C:\Users\cida>
```

Docker – Removendo *container*

```
root@30349ad0263a: /home x Windows PowerShell x + v
PS C:\Users\cida> docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
30349ad0263a   ubuntu   "bash"    10 minutes ago   Up 10 minutes           gallant_keller
PS C:\Users\cida>
PS C:\Users\cida> docker rm 30349ad0263a
Error response from daemon: You cannot remove a running container 30349ad0263a0c01b63dfa786a57b43831ce3eba993cbd4eb13a5be9cd4b063d. Stop the container before attempting removal or force remove
PS C:\Users\cida> |
```

```
PS C:\Users\cida>
PS C:\Users\cida> docker stop -t=0 30349ad0263a
30349ad0263a
PS C:\Users\cida> |
```

```
# -t = 0 stop imediatamente
$ docker stop -t=0 30349ad0263a
```

■ Docker – Removendo *container*

- Voltar no terminal onde o Ubuntu estava rodando e veja que não estamos mais no *container*.

```
PS C:\Users\cida> docker run -it ubuntu bash
root@30349ad0263a:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
root@30349ad0263a:/# cd home
root@30349ad0263a:/home# ls
ubuntu
root@30349ad0263a:/home#
PS C:\Users\cida> |
```


Docker

- Vamos executar o container novamente.

```
PS C:\Users\cida> docker run -it ubuntu bash
root@5e7eeef3ae8a:/# ls
bin    dev    home  lib64  mnt    proc   run    srv    tmp    var
boot  etc    lib   media  opt    root   sbin   sys    usr
root@5e7eeef3ae8a:/# |
```

- No outro terminal vamos listar os *containers*. Veja que o *id* e o *name* do container mudaram.

```
root@5e7eeef3ae8a: /  Windows PowerShell
PS C:\Users\cida> docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
5e7eeef3ae8a   ubuntu   "bash"    About a minute ago   Up About a minute   dreamy_curie
```

Docker

- Vamos *forçar* para remover o *container*. `--force` para e remove o *container*

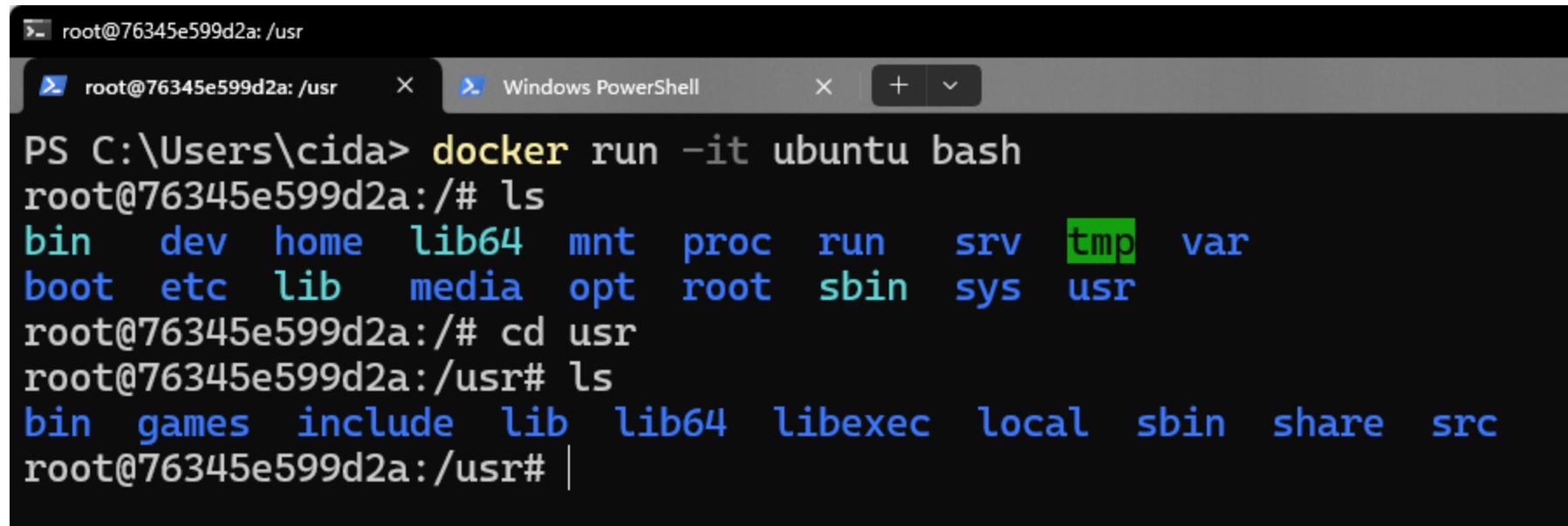
```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\cida> docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
5e7eeef3ae8a   ubuntu   "bash"    About a minute ago   Up About a minute     dreamy_curie
PS C:\Users\cida> docker rm 5e7eeef3ae8a --force
5e7eeef3ae8a
PS C:\Users\cida> |
```

- Veja no outro terminal que o *container* foi encerrado.

```
PS C:\Users\cida> docker run -it ubuntu bash
root@5e7eeef3ae8a:/# ls
bin    dev    home  lib64  mnt    proc   run    srv    tmp    var
boot  etc    lib   media  opt    root   sbin   sys    usr
root@5e7eeef3ae8a:/#
PS C:\Users\cida> |
```

Docker – Subindo e parando o container

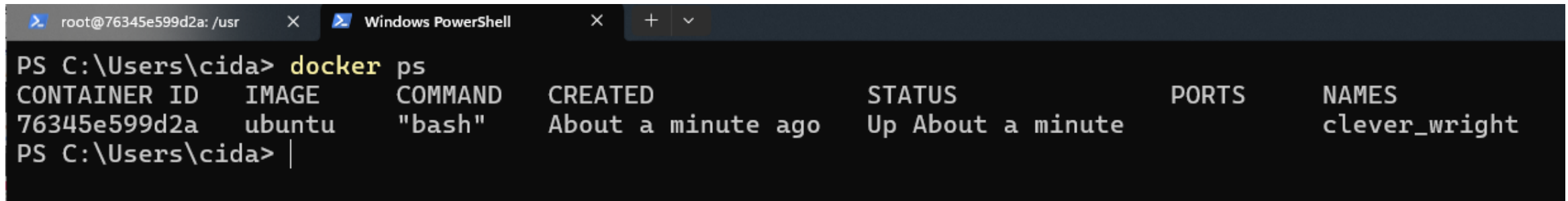
- Vamos executar novamente o *container* do Ubuntu.



```
root@76345e599d2a: /usr
root@76345e599d2a: /usr
Windows PowerShell
PS C:\Users\cida> docker run -it ubuntu bash
root@76345e599d2a:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
root@76345e599d2a:/# cd usr
root@76345e599d2a:/usr# ls
bin  games  include  lib  lib64  libexec  local  sbin  share  src
root@76345e599d2a:/usr# |
```

Docker

- Abrir outro terminal.



The screenshot shows a Windows PowerShell terminal window with two tabs: 'root@76345e599d2a: /usr' and 'Windows PowerShell'. The active tab is 'Windows PowerShell', showing the command 'docker ps' and its output. The output is a table with columns: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, PORTS, and NAMES. The table contains one row for container '76345e599d2a' with image 'ubuntu' and command '"bash"'. The status is 'Up About a minute'.

```
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
76345e599d2a   ubuntu   "bash"    About a minute ago   Up About a minute   clever_wright
PS C:\Users\cida> |
```

Parar containers

`docker stop <id ou name>`, para parar a execução do **container** e liberar recursos por ele utilizados.

Iniciar/Reiniciar containers

`docker start <id ou name>` volta a execução de um container.

O comando `run` **sempre cria um container novo**.

```
$ docker stop <id ou name>
# ou
$ docker container stop <id ou name do container>
```

```
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
76345e599d2a   ubuntu   "bash"    About a minute ago   Up About a minute   clever_wright
PS C:\Users\cida> docker stop 763
763
PS C:\Users\cida> docker start 763
763
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
76345e599d2a   ubuntu   "bash"    38 minutes ago   Up 10 seconds        clever_wright
PS C:\Users\cida> docker container stop clever_wright
clever_wright
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Users\cida> docker container start clever_wright
clever_wright
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
76345e599d2a   ubuntu   "bash"    41 minutes ago   Up 3 seconds        clever_wright
PS C:\Users\cida> |
```

Da mesma forma como podemos utilizar o `stop` e `start` para parar/iniciar um **container**, também podemos reiniciá-lo com o comando `restart`.

```
$ docker container restart <CONTAINER ID>
```

Para acessar o **container** em execução novamente utilizamos o parâmetro `attach` do comando `docker container`

```
$ docker container attach <CONTAINER ID>
```

```
PS C:\Users\cida>
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
a8ab1e66da4c   ubuntu   "bash"    14 seconds ago   Up 14 seconds           mystifying_rosalind
PS C:\Users\cida> docker stop a8a
a8a
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Users\cida> docker restart a8a
a8a
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
a8ab1e66da4c   ubuntu   "bash"    53 seconds ago   Up 2 seconds           mystifying_rosalind
PS C:\Users\cida> docker container attach a8a
root@a8ab1e66da4c:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@a8ab1e66da4c:/# |
```


Docker – Modo *detached*

Vamos entender o comando a seguir:

```
$ docker run -d -p 80:80 docker/getting-started
```

1. `docker run` :
 - Inicia um novo **container** com base em uma imagem.
 - O comando `docker run` é usado para criar e executar um **container** a partir de uma imagem.
2. `-d (Modo Detached)`:
 - Executa o **container** em segundo plano (modo desanexado).
 - Isso significa que o **container** continuará sendo executado mesmo após você fechar o terminal ou a janela do console.
3. `-p 80:80` :
 - Mapeia a porta **80** do *host* (computador) para a porta **80** do **container**.
 - Isso permite que o tráfego enviado para a porta **80** do *host* seja redirecionado para a porta **80** do **container**.
4. `docker/getting-started` :
 - Especifica a imagem a ser usada para criar o **container**. **docker** é o usuário no Docker Hub.
 - Neste caso, o **container** será criado a partir da imagem chamada `docker/getting-started`.

Resumindo: o comando inicia um **container** a partir da imagem `docker/getting-started`, executa-o em segundo plano e mapeia a porta **80** do *host* para a porta **80** do **container**. Isso permitirá que você acesse o **container** via HTTP no endereço `http://localhost:80` pelo navegador.

```
Windows PowerShell
Windows PowerShell
PS C:\Users\cida> docker run -d -p 80:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
c158987b0551: Pull complete
1e35f6679fab: Pull complete
cb9626c74200: Pull complete
b6334b6ace34: Pull complete
f1d1c9928c82: Pull complete
9b6f639ec6ea: Pull complete
ee68d3549ec8: Pull complete
33e0cbbb4673: Pull complete
4f7e34c2de10: Pull complete
Digest: sha256:d79336f4812b6547a53e735480dde67f8f8f7071b414fbd9297609ffb989abc1
Status: Downloaded newer image for docker/getting-started:latest
00183f5d440677199f49b651159235063642fbc93991b15dc9643da959ee324a
PS C:\Users\cida> |
```

- Abrir o navegador – localhost:80

The screenshot shows a web browser window with the address bar at localhost/tutorial/. The browser's tab bar shows various open tabs including API, Imagens e Sons Gr..., Ferramentas, Facebook, Google & YT, Klick X, Canal, Microsoft, Educação, Cursos, Docker, .NET, Hospital, PGAdmin, github, Azure, Importado, and D. The page header is blue with the Docker Labs logo, the title 'Getting Started', a search bar, and a GitHub repository link for 'docker/getting-started' with 2.8k Stars and 6.4k Forks. The main content area has a left sidebar with a table of contents, a central 'Getting Started' section, and a right sidebar with a 'Table of contents' section. The central section is titled 'Getting Started' and 'The command you just ran'. It contains a code block with the command 'docker run -d -p 80:80 docker/getting-started' and a list of flags: '-d' (detached mode), '-p 80:80' (port mapping), and 'docker/getting-started' (image name). Below this is a 'Pro tip' section explaining that single character flags can be combined to shorten the command to 'docker run -dp 80:80 docker/getting-started'.

localhost/tutorial/

API Imagens e Sons Gr... Ferramentas Facebook Google & YT Klick X Canal Microsoft Educação Cursos Docker .NET Hospital PGAdmin github Azure Importado D

docker Labs Getting Started Search docker/getting-started 2.8k Stars - 6.4k Forks

Getting Started
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?

Getting Started

The command you just ran

Congratulations! You have started the container for this tutorial! Let's first explain the command that you just ran. In case you forgot, here's the command:

```
docker run -d -p 80:80 docker/getting-started
```

You'll notice a few flags being used. Here's some more info on them:

- `-d` - run the container in detached mode (in the background)
- `-p 80:80` - map port 80 of the host to port 80 in the container
- `docker/getting-started` - the image to use

Pro tip

You can combine single character flags to shorten the full command. As an example, the command above could be written as:

```
docker run -dp 80:80 docker/getting-started
```

Table of contents
The command you just ran
The Docker Dashboard
What is a container?
What is a container image?

```
Windows PowerShell
Windows PowerShell
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
NAMES
00183f5d4406   docker/getting-started "/docker-entrypoint...." 51 seconds ago Up 49 seconds 0.0.0.0:80->80/tcp
strange_kare
PS C:\Users\cida> |
```

Docker

Containers docker não tem conexão com nada externo a eles.

A flag `-p` `<porta_local_de_acesso>:<porta_do_container>` expõe portas para que o **container** fique acessível.

`docker logs -f <id ou name>`, com a flag `-f` (*follow*) mostra os logs em tempo real.

Digite `ctrl+c` para sair do log.

```
CONTAINER ID   IMAGE                                COMMAND                                            CREATED        STATUS        PORTS
NAMES
00183f5d4406   docker/getting-started              "/docker-entrypoint...."                        51 seconds ago Up 49 seconds  0.0.0.0:80->80/tcp
strange_kare
PS C:\Users\cida> docker logs -f 001
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/05/07 21:55:06 [notice] 1#1: using the "epoll" event method
2024/05/07 21:55:06 [notice] 1#1: nginx/1.23.3
2024/05/07 21:55:06 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1_git20220924-r4)
2024/05/07 21:55:06 [notice] 1#1: OS: Linux 5.15.90.1-microsoft-standard-WSL2
2024/05/07 21:55:06 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/05/07 21:55:06 [notice] 1#1: start worker processes
2024/05/07 21:55:06 [notice] 1#1: start worker process 30
```

- Analise o *log* gerado. Vá para a última linha do *log* e recarregue a página no navegador.

Docker – Acessando o *container*

```
$ docker exec -it <id ou name> bash (ou sh, depende da imagem)
# ou
$ docker container exec -it <id ou name> bash (ou sh, depende da imagem)
```

A *flag* `-it` possibilita executar comandos disponíveis no ***container*** por meio de terminal interativo

`-t` - Disponibiliza um TTY (console) para o container.

`-i` - Mantém o STDIN aberto mesmo que você não esteja conectado no container.

Para sair do modo interativo digite **CTRL+D**

Docker – Acessando o *container*

```
PS C:\Users\cida> docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
NAMES
00183f5d4406   docker/getting-started "/docker-entrypoint..." 5 minutes ago   Up 5 minutes   0.0.0.0:80->80/tcp
strange_kare
PS C:\Users\cida> docker exec -it 001 /bin/sh
/ # ls
bin          etc          mnt          run          tmp
dev          home         opt          sbin         usr
docker-entrypoint.d  lib         proc         srv          var
docker-entrypoint.sh media        root         sys

/ # ps -ef
PID   USER     TIME   COMMAND
    1  root      0:00   nginx: master process nginx -g daemon off;
   30  nginx    0:00   nginx: worker process
   31  nginx    0:00   nginx: worker process
   32  nginx    0:00   nginx: worker process
   33  nginx    0:00   nginx: worker process
   34  nginx    0:00   nginx: worker process
   35  nginx    0:00   nginx: worker process
   36  nginx    0:00   nginx: worker process
   37  nginx    0:00   nginx: worker process
   38  nginx    0:00   nginx: worker process
   39  nginx    0:00   nginx: worker process
   40  nginx    0:00   nginx: worker process
   41  nginx    0:00   nginx: worker process
   42  root      0:00   /bin/sh
   49  root      0:00   ps -ef

/ #
```




Copyright © 2024
Prof^a. Aparecida de Fátima Castello Rosa

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).