

# Systems Advanced Docker Containers

---

Containers runnen met  
docker CLI



Elfde-Liniestraat 24, 3500 Hasselt, [www.pxl.be](http://www.pxl.be)



# Een commando meegeven aan docker run

Indien we een commando meegeven aan de Ubuntu container, zal deze dit commando uitvoeren en afsluiten.

De ubuntu container doet standaard niets. **Een docker container is een eigenlijk een gewoon linux proces** en als dat proces niets te doen heeft exit het onmiddellijk.

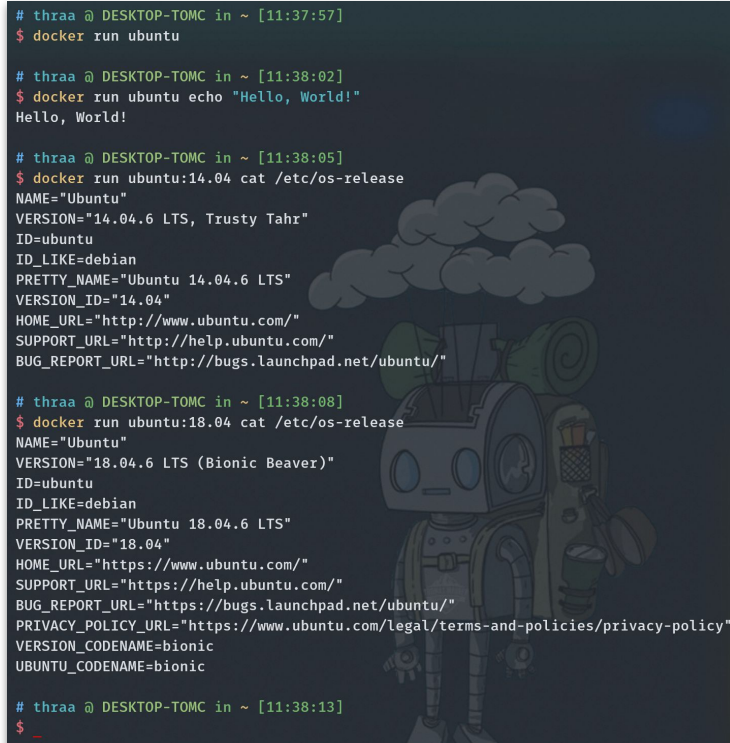
`docker run ubuntu`

We kunnen wel een commando meegeven, zodat dit uitgevoerd wordt.

`docker run ubuntu echo "Hello, World!"`

`docker run ubuntu:14.04 cat /etc/os-release`

`docker run ubuntu:18.04 cat /etc/os-release`

A terminal window with a dark background. On the right side, there is a stylized illustration of a robot with a cylindrical body, a helmet, and a backpack. The robot is standing and looking towards the left. The terminal text shows a series of commands and their outputs. The first command is 'docker run ubuntu', which runs without output. The second command is 'docker run ubuntu echo "Hello, World!"', which outputs 'Hello, World!'. The third command is 'docker run ubuntu:14.04 cat /etc/os-release', which outputs system information for Ubuntu 14.04.6 LTS. The fourth command is 'docker run ubuntu:18.04 cat /etc/os-release', which outputs system information for Ubuntu 18.04.6 LTS (Bionic Beaver).

```
# thraa @ DESKTOP-TOMC in ~ [11:37:57]
$ docker run ubuntu

# thraa @ DESKTOP-TOMC in ~ [11:38:02]
$ docker run ubuntu echo "Hello, World!"
Hello, World!

# thraa @ DESKTOP-TOMC in ~ [11:38:05]
$ docker run ubuntu:14.04 cat /etc/os-release
NAME="Ubuntu"
VERSION="14.04.6 LTS, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 14.04.6 LTS"
VERSION_ID="14.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"

# thraa @ DESKTOP-TOMC in ~ [11:38:08]
$ docker run ubuntu:18.04 cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.6 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.6 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic

# thraa @ DESKTOP-TOMC in ~ [11:38:13]
$
```

# Een container detached opstarten

Wanneer we een container gewoon met `docker run` opstarten, krijgen we onze prompt niet terug.

Bv, we voeren een `sleep` commando uit en we krijgen de prompt niet terug zolang de container hiermee bezig is (process id 1).

```
docker run ubuntu sleep 7
```

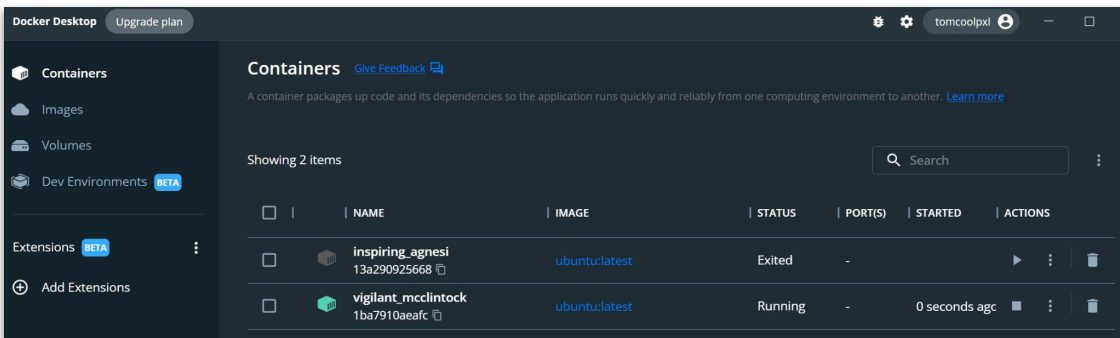
Nu doen we een `sleep`, maar we starten de container op in **detached** mode, zodat we onze prompt onmiddellijk terugkrijgen. De container draait terwijl verder in de **achtergrond**.

```
docker run -d ubuntu sleep 20
```

```
# thraa @ DESKTOP-TOMC in ~ [11:42:02]
$ docker run ubuntu sleep 7

# thraa @ DESKTOP-TOMC in ~ [11:42:31]
$ docker run -d ubuntu sleep 20
1ba7910aeafcb8541bf9d1c0108e1b5e069cc044c3a86d0f77fbf64bf54fc7d5

# thraa @ DESKTOP-TOMC in ~ [11:42:44]
$ _
```

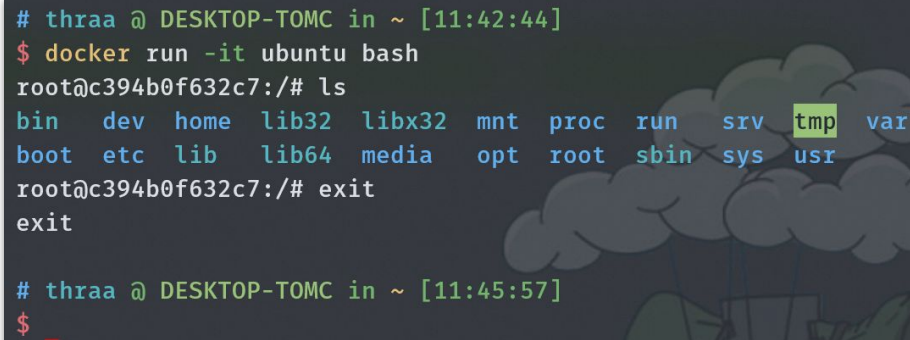


# Een interactieve sessie starten

Indien we een shell willen in de container, starten we een **i**nteractieve **t**ty-sessie.

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

```
docker run -it ubuntu /bin/bash
```

A terminal window with a dark background and a cartoon illustration of a person in a green shirt and hat on the right. The terminal shows a user named 'thraa' at a machine named 'DESKTOP-TOMC' starting a Docker container. The container is an Ubuntu shell. The user runs 'ls' and sees a list of directories including 'tmp' (highlighted in green). The user then runs 'exit' and returns to the host shell.

```
# thraa @ DESKTOP-TOMC in ~ [11:42:44]
$ docker run -it ubuntu bash
root@c394b0f632c7:/# ls
bin    dev    home  lib32  libx32  mnt    proc  run    srv    tmp    var
boot   etc    lib   lib64  media   opt    root  sbin   sys    usr
root@c394b0f632c7:/# exit
exit

# thraa @ DESKTOP-TOMC in ~ [11:45:57]
$ _
```

We kunnen de interactieve sessie afsluiten met het **exit** commando: dan stopt het proces (op pid 1) en stopt de container.

Indien je de **-t** vergeet krijg je geen prompt, maar kan je nog altijd commando's geven en krijg je output.

- Je kan dan de container verlaten met **exit** (of in de shell met [CTRL]+[D])

Indien je de **-i** vergeet krijg je een prompt, maar kan je geen commando's geven.

- Je kan de container dan killen vanuit een andere tty met **docker kill <CONTAINER>**.

# Een interactieve sessie starten

Indien we een shell willen in de container, starten we een interactieve **t**ty-sessie.

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

Je kan de container verlaten zonder die af te sluiten met de volgende toetsencombinatie:




Je kan opnieuw de interactieve shell verkrijgen met

```
docker attach <CONTAINER>
```

waar <CONTAINER> de container id is of de container funky name

```
# thraa @ DESKTOP-TOMC in ~ [11:52:36]
$ docker run -it ubuntu bash
root@282c21e7fc3d:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@282c21e7fc3d:/# exit
exit

# thraa @ DESKTOP-TOMC in ~ [11:52:50]
$ docker run -it ubuntu bash
root@e3660bf3c0f3:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@e3660bf3c0f3:/# 
exit

# thraa @ DESKTOP-TOMC in ~ [11:53:15]
$ docker attach e3660bf3c0f3
root@e3660bf3c0f3:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@e3660bf3c0f3:/# exit
exit

# thraa @ DESKTOP-TOMC in ~ [11:53:57]
$ _
```

# Een overzicht van containers

We kunnen een overzicht van de containers krijgen met:

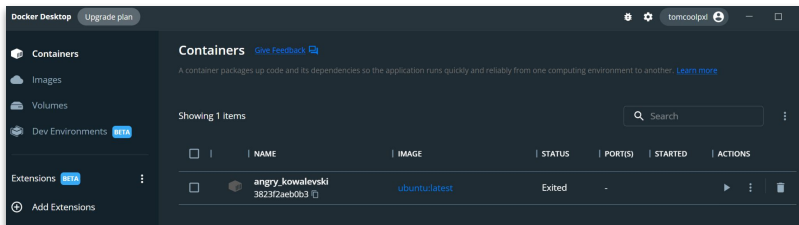
`docker container ls` of `docker ps`

Overzicht van alle containers:

`docker container ls -a`

Een container verwijderen:

`docker container rm <CONTAINER>`



# Container (re)naming

We kunnen een container een naam geven tijdens het opstarten, of later hernoemen.

Een container een naam geven bij het aanmaken:

```
docker run --name <name> <image>
```

Een container een andere naam geven:

```
docker container rename <CONTAINER>  
<NEW_NAME>
```

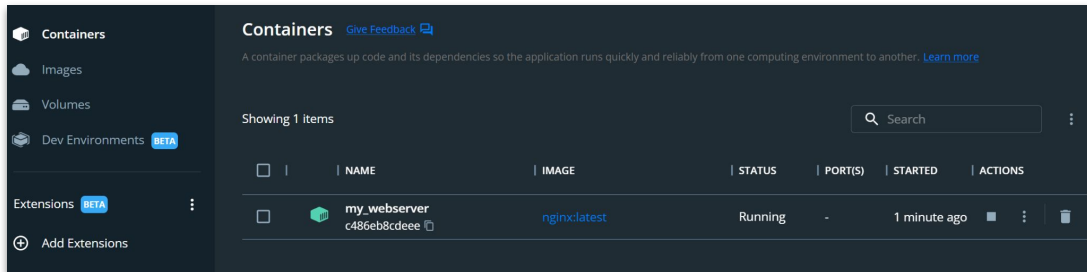
```
# thraa @ DESKTOP-TOMC in ~ [12:17:52]
$ docker run -d --name my-container nginx:latest
c486eb8cdeee110f68bf2f20a1d9b5e421cc711e46446552f3706175133fb98c

# thraa @ DESKTOP-TOMC in ~ [12:17:56]
$ docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
c486eb8cdeee   nginx:latest   "/docker-entrypoint..." 4 seconds ago  Up 4 seconds  80/tcp       my-container

# thraa @ DESKTOP-TOMC in ~ [12:18:01]
$ docker container rename my-container my_webserver

# thraa @ DESKTOP-TOMC in ~ [12:18:10]
$ docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
c486eb8cdeee   nginx:latest   "/docker-entrypoint..." 16 seconds ago Up 15 seconds  80/tcp       my_webserver

# thraa @ DESKTOP-TOMC in ~ [12:18:12]
$ _
```





# Container states

We kunnen een container pauzeren.

Een container pauzeren:

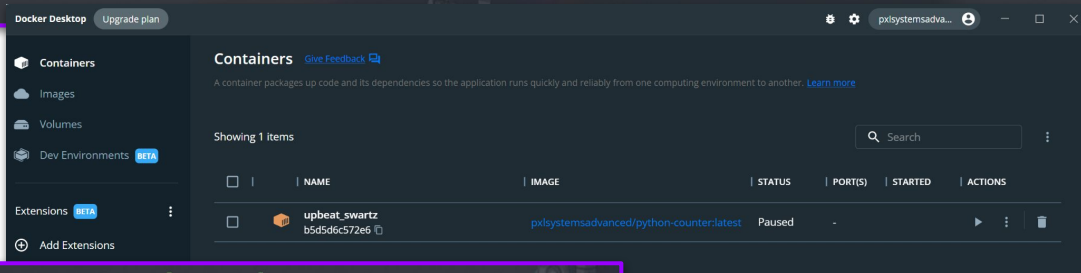
```
docker container pause  
<CONTAINER>
```

Een container laten verder gaan:

```
docker container unpause  
<CONTAINER>
```

```
# thraa @ DESKTOP-TOMC in ~ [23:02:15]  
$ docker run pxlsystemsadvanced/python-counter  
1  
2  
3  
4  
—
```

```
# thraa @ DESKTOP-TOMC in ~ [23:01:48]  
$ docker container pause upbeat_swartz  
upbeat_swartz  
  
# thraa @ DESKTOP-TOMC in ~ [23:02:29]  
$ docker container ls  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS          NAMES  
b5d5d6c572e6   pxlsystemsadvanced/python-counter   "python -u ./counter..." About a minute ago Up About a minute (Paused)          upbeat_swartz  
  
# thraa @ DESKTOP-TOMC in ~ [23:03:45]  
$ —
```



```
# thraa @ DESKTOP-TOMC in ~ [23:03:45]  
$ docker container unpause upbeat_swartz  
upbeat_swartz  
  
# thraa @ DESKTOP-TOMC in ~ [23:05:28]  
$ —
```

5  
6  
7  
8  
9  
10  
11  
12  
13



# Container states

stoppen/starten van een container.

- Een container stoppen:
  - `docker container stop <CONTAINER>`

We kunnen ipv `stop` ook `kill` gebruiken

- `stop`: send SIGTERM, and then SIGKILL after grace period
- `kill`: send SIGKILL
- Een container starten:
  - `docker container start <CONTAINER>`

We kunnen een container ook opnieuw **i**nteractief en **a**ttached starten met de opties `-ia`

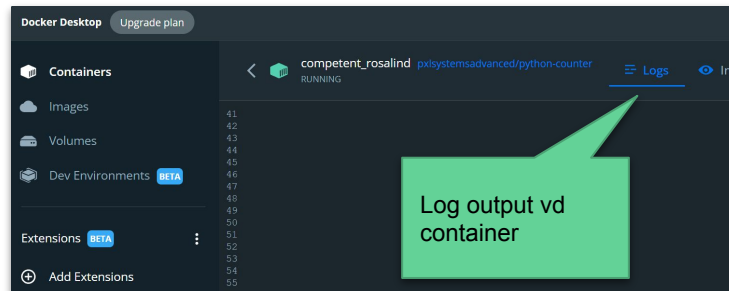
We kunnen ipv `start` ook `restart` gebruiken om opnieuw op te starten.

```
# thraa @ DESKTOP-TOMC in ~ [23:13:47]
$ docker run pxlsystemsadvanced/python-counter
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

SIGTERM...

SIGKILL

```
# thraa @ DESKTOP-TOMC in ~ [23:14:20] C:137
$ -
```



```
# thraa @ DESKTOP-TOMC in ~ [23:13:44]
$ docker container stop competent_rosalind
competent_rosalind

# thraa @ DESKTOP-TOMC in ~ [23:14:20]
$ docker container start competent_rosalind
competent_rosalind

# thraa @ DESKTOP-TOMC in ~ [23:17:10]
$ docker container logs competent_rosalind
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
```

Hier is de container opnieuw gestart

Om interactief te kijken moeten we opnieuw attachen

```
# thraa @ DESKTOP-TOMC in ~ [23:17:15]
$ docker container attach competent_rosalind
26
27
```

# Container logs & -stats

We kunnen wat meer info bekomen met **stats** en **logs**.

**docker container stats**  
**<CONTAINER>**

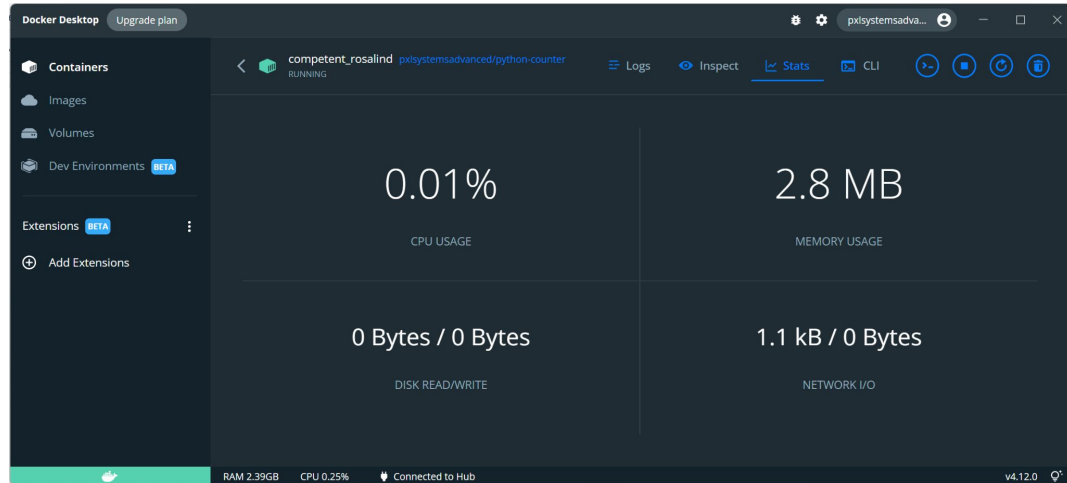
**docker container logs**  
**<CONTAINER>**

Je kan de logs ook blijven bekijken met  
**docker container logs -f**

```
# thraa @ DESKTOP-TOMC in ~ [23:24:03] C:1
$ docker container stats competent_rosalind

# thraa @ DESKTOP-TOMC in ~ [23:24:05] C:1
$ docker container stats competent_rosalind --no-stream
CONTAINER ID   NAME                CPU %     MEM USAGE / LIMIT   MEM %     NET I/O       BLOCK I/O   PIDS
812d11f10366   competent_rosalind  0.01%    2.82MiB / 15.59GiB   0.02%    1.02kB / 0B    0B / 0B     1

# thraa @ DESKTOP-TOMC in ~ [23:24:13]
$ docker container logs competent_rosalind
1
2
3
4
```



# Containers verwijderen

**docker container prune** om alle gestopte containers ineens te verwijderen.

```
$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
812d11f10366   pxlsystemsadvanced/python-counter   "python -u ./counter..." 13 minutes ago Up 4 minutes                competent_rosalind

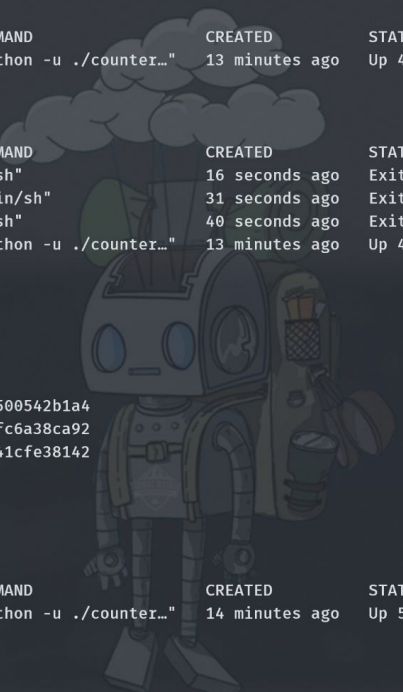
# thraa @ DESKTOP-TOMC in ~ [23:27:37]
$ docker container ls --all
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
cd97d7eb5856   ubuntu                              "bash"                 16 seconds ago Exited (0) 14 seconds ago          wizardly_hopper
b6468d889b9a   alpine                              "/bin/sh"              31 seconds ago Exited (0) 30 seconds ago          friendly_payne
9b2949dbdf53   ubuntu:18.04                       "bash"                 40 seconds ago Exited (0) 38 seconds ago          vibrant_kowalevski
812d11f10366   pxlsystemsadvanced/python-counter   "python -u ./counter..." 13 minutes ago Up 4 minutes                competent_rosalind

# thraa @ DESKTOP-TOMC in ~ [23:27:45]
$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
cd97d7eb5856847750f4b417ab982a78c172b5d89dde4c06109e80500542b1a4
b6468d889b9aea605a2a383e08f99fbb785411eb47171e4db941e4fc6a38ca92
9b2949dbdf53d5fb46c73d8aee1766294da79e27517abe0900b9cb41cfe38142

Total reclaimed space: 0B

# thraa @ DESKTOP-TOMC in ~ [23:28:11]
$ docker container ls --all
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
812d11f10366   pxlsystemsadvanced/python-counter   "python -u ./counter..." 14 minutes ago Up 5 minutes                competent_rosalind

# thraa @ DESKTOP-TOMC in ~ [23:28:15]
$
```



# Containers verwijderen

Gestopte container  
verwijderen:

```
docker container rm  
<CONTAINER>
```

```
# thraa @ DESKTOP-TOMC in ~ [23:30:33]
$ docker container ls --all
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
# thraa @ DESKTOP-TOMC in ~ [23:30:35]
$ docker run -d --name my-counter_bg pxlsystemsadvanced/python-counter
f1fded1598f5d2482b54a1b67b16b5afec452d9529faf3b3e6140b05822bde6a
# thraa @ DESKTOP-TOMC in ~ [23:30:47]
$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
f1fded1598f5   pxlsystemsadvanced/python-counter   "python -u ./counter..." 7 seconds ago   Up 6 seconds          my-counter_bg
# thraa @ DESKTOP-TOMC in ~ [23:30:54]
$ docker container rm my-counter_bg
Error response from daemon: You cannot remove a running container f1fded1598f5d2482b54a1b67b16b5afec452d9529faf3b3e6140b05822bde6a. Stop the container
before attempting removal or force remove
# thraa @ DESKTOP-TOMC in ~ [23:31:29] C:1
$ docker container kill my-counter_bg
my-counter_bg
# thraa @ DESKTOP-TOMC in ~ [23:31:40]
$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  STATUS        PORTS        NAMES
# thraa @ DESKTOP-TOMC in ~ [23:31:42]
$ docker container ls --all
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
f1fded1598f5   pxlsystemsadvanced/python-counter   "python -u ./counter..." About a minute ago   Exited (137) 10 seconds ago   my-counter_bg
# thraa @ DESKTOP-TOMC in ~ [23:31:50]
$ docker container rm my-counter_bg
my-counter_bg
# thraa @ DESKTOP-TOMC in ~ [23:32:04]
$ docker container ls --all
CONTAINER ID   IMAGE                                COMMAND                  STATUS        PORTS        NAMES
```

# Containers verwijderen

Om een running container te verwijderen: `docker container rm --force <CONTAINER>`

```
# thraa @ DESKTOP-TOMC in ~ [23:34:05]
$ docker container ls --all
CONTAINER ID    IMAGE                COMMAND             CREATED        STATUS        PORTS        NAMES

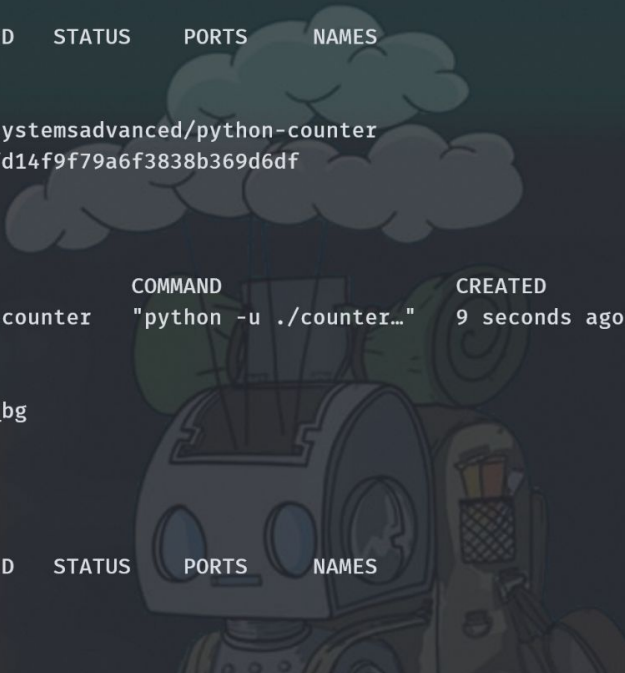
# thraa @ DESKTOP-TOMC in ~ [23:34:06]
$ docker run -d --name my-counter_bg pxlsystemsadvanced/python-counter
b22a95b351fe1842476ef6a36845128b5de333d9fd14f9f79a6f3838b369d6df

# thraa @ DESKTOP-TOMC in ~ [23:34:10]
$ docker container ls
CONTAINER ID    IMAGE                COMMAND             CREATED        STATUS        PORTS        NAMES
b22a95b351fe    pxlsystemsadvanced/python-counter    "python -u ./counter..."    9 seconds ago    Up 7 seconds                my-counter_bg

# thraa @ DESKTOP-TOMC in ~ [23:34:18]
$ docker container rm --force my-counter_bg
my-counter_bg

# thraa @ DESKTOP-TOMC in ~ [23:34:30]
$ docker container ls --all
CONTAINER ID    IMAGE                COMMAND             CREATED        STATUS        PORTS        NAMES

# thraa @ DESKTOP-TOMC in ~ [23:34:36]
$
```



# Oefening: werken met images

## Verschillende distributies op één Docker-host

Run volgende containers (van Docker-hub)

- Ubuntu 22.04
- Fedora (latest)
- Arch linux (latest)

en probeer om telkens volgende info te tonen door in de containers te kruipen

- de distributie-informatie
- de shell die er gedraaid wordt
- kernel-info

# Oefening: werken met containers

## Zorg voor een interactieve sessie in een Oracle Linux-container (laatste versie)

- geef de container de naam: OLContainer
- laat zien dat je daadwerkelijk in een Oracle Linux-container zit door de versie van Oracle Linux te tonen
- installeer tree
- maak met nano een bestand aan in je homedirectory (met een willekeurige tekst om te testen)
- verlaat de container zonder hem te stoppen
- re-attach naar de container
- verlaat de container zonder hem te stoppen
- pauzeer de container
- stop de container
- start de container
- reattach naar de container
- controleer of het bestand in je homedirectory nog bestaat
- verlaat de container zodat deze ook gestopt is
- verwijder de container

Voor hulp:

`docker --help`

Voor specifieke hulp:

`docker <command> --help`

Kijk ook zeker eens op

<https://docs.docker.com>

→ Reference, Docker CLI



# Oefening: werken met containers

- Start drie Oracle Linux-containers op die onmiddellijk stoppen met als namen:
  - OLstopped1
  - OLstopped2
  - OLstopped3
- Start drie OL-containers op die in de achtergrond blijven draaien met als namen:
  - OLrunning1
  - OLrunning2
  - OLrunning3
- Toon enkel de Container-short-ID van alle runnende containers
- Toon enkel de Container-short-ID van alle containers, ook die op de achtergrond draaien
- Verwijder de container OLstopped1
- Verwijder de op de achtergrond runnende container OLrunning1 zonder deze eerst te stoppen
- Verwijder ineens alle gestopte containers
- Verwijder ineens alle draaiende containers zonder deze eerst te stoppen
  - TIP: gebruik hiervoor de uitkomst van de derde vraag in een subshell
- Start een nieuwe gestopte container OLstopped4 en een op de achtergrond draaiende OLrunning4
- Verwijder ineens alle containers (gestopt en draaiend)

Voor hulp:

`docker --help`

Voor specifieke hulp:

`docker <command> --help`

Kijk ook zeker eens op

<https://docs.docker.com>

→ Reference, Docker CLI

