

# Operating Systems

---

## Praktijk - Installatie Docker binnen Fedora

---

### Doel

---

Tijdens dit labo zullen we Docker installeren bovenop de Fedora VM, en enkele eenvoudige containers opstarten.

Na dit labo kan je:

- Docker installeren op een toestel met Linux
- Een bestaande image importeren en opstarten in een nieuwe container
- De lokale containers en images beheren

### Installatie Docker

---

De meeste Linux distributies hebben een ingebouwde package manager (bv. *dnf* bij Fedora, *apt* bij Ubuntu) die toelaten om op een eenvoudige manier de Docker Engine te installeren. Vaak bevatten de standaard repositories die ingebouwd zijn in de gekozen Linux distributie echter een verouderde versie van Docker. Daarom kiezen we er hier voor om de meest recente stabiele (*stable*) versie rechtstreeks te installeren via de officiële Docker repository. Voor de installatie baseren we ons op de officiële installatiehandleiding, die te vinden is op <https://docs.docker.com/engine/install/fedora/>.

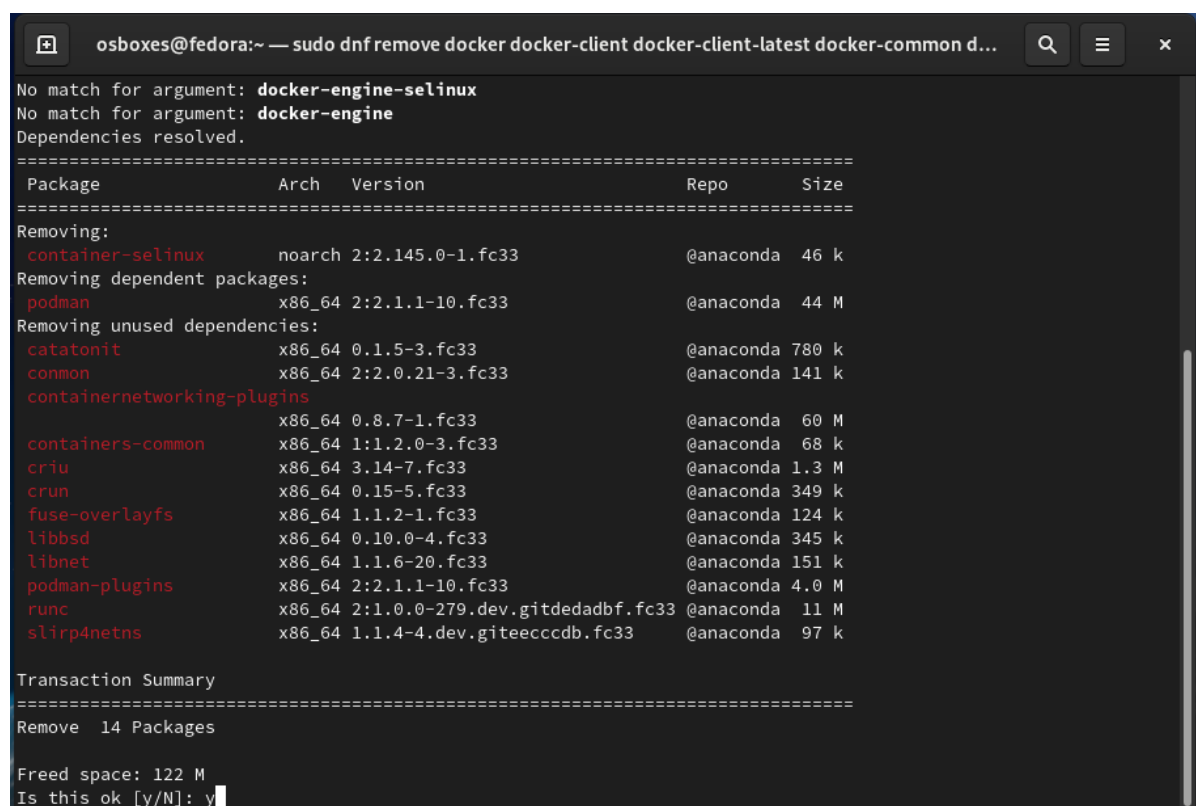
Docker maakt gebruik van kernel-virtualisatie, en de Docker engine draait dus bovenop een bestaande Linux kernel. Het is mogelijk om Docker rechtstreeks te installeren in Windows, maar hiervoor wordt op de achtergrond een virtuele machine met Linux gebruikt die draait bovenop Hyper-V. Er bestaat echter ook een variant, Docker Toolbox, die kan draaien bovenop VirtualBox, maar dit is een oudere versie van Docker. We beschikken echter reeds over een virtuele machine met Fedora, dus voor de eenvoud kiezen we er voor om Docker te installeren bovenop deze virtuele machine.

### Oudere versies verwijderen

Het is mogelijk om Docker te installeren via de default repositories van Fedora. Dit is echter een verouderde versie, en in deze opdracht kiezen we er voor om te werken met de laatste stabiele release. Mocht je echter reeds een oudere versie van Docker geïnstalleerd hebben, kan je deze eenvoudig verwijderen (samen met alle dependencies) via onderstaand commando:

```
sudo dnf remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-selinux \
    docker-engine-selinux \
    docker-engine
```

Ook als je nog geen Docker gebruikt hebt op de VM, is het een goed idee om bovenstaand commando uit te voeren in een terminal. Bij uitvoeren van dit commando zal je merken dat veel van deze packages (nog) niet geïnstalleerd zijn, maar enkele packages (waaronder *container-selinux* en *podman*) zal je wel kunnen verwijderen. Bevestig het verwijderen door **y** in te geven na de prompt, zoals te zien in onderstaande afbeelding



```
osboxes@fedora:~ — sudo dnf remove docker docker-client docker-client-latest docker-common d...
No match for argument: docker-engine-selinux
No match for argument: docker-engine
Dependencies resolved.
=====
Package                Arch  Version                Repo                Size
=====
Removing:
container-selinux      noarch 2:2.145.0-1.fc33      @anaconda          46 k
Removing dependent packages:
podman                 x86_64 2:2.1.1-10.fc33      @anaconda          44 M
Removing unused dependencies:
catatonit              x86_64 0.1.5-3.fc33         @anaconda          780 k
common                 x86_64 2:2.0.21-3.fc33      @anaconda          141 k
containernetworking-plugins
x86_64                 0.8.7-1.fc33         @anaconda          60 M
containers-common      x86_64 1:1.2.0-3.fc33        @anaconda          68 k
criu                   x86_64 3.14-7.fc33           @anaconda          1.3 M
crun                   x86_64 0.15-5.fc33           @anaconda          349 k
fuse-overlayfs         x86_64 1.1.2-1.fc33         @anaconda          124 k
libbsd                 x86_64 0.10.0-4.fc33        @anaconda          345 k
libnet                 x86_64 1.1.6-20.fc33        @anaconda          151 k
podman-plugins         x86_64 2:2.1.1-10.fc33      @anaconda          4.0 M
runc                   x86_64 2:1.0.0-279.dev.gitdedadbf.fc33 @anaconda          11 M
slirp4netns            x86_64 1.1.4-4.dev.giteeccdb.fc33 @anaconda          97 k

Transaction Summary
=====
Remove  14 Packages

Freed space: 122 M
Is this ok [y/N]: y
```

## Toevoegen repository

Om de laatste versie van Docker te installeren, zullen we de officiële Docker repository toevoegen aan de *dnf* package manager die gebruikt wordt binnen Fedora. Hiervoor moeten we eerst de *dnf-plugins-core* package installeren, waarna we een nieuwe repository kunnen toevoegen:

```
sudo dnf -y install dnf-plugins-core
```

```
sudo dnf config-manager \
    --add-repo \
    https://download.docker.com/linux/fedora/docker-ce.repo
```

# Installatie Docker Engine

Nadat we de officiële Docker repository toegevoegd hebben aan *dnf* kunnen we eenvoudig de laatste stabiele versie van Docker installeren in Fedora:

```
sudo dnf install docker-ce docker-ce-cli containerd.io
```

Je krijgt een overzicht te zien van alle software die geïnstalleerd zal worden, en je kan de installatie bevestigen door opnieuw **y** in te geven na de prompt.

```
osboxes@fedora:~ — sudo dnf install docker-ce docker-ce-cli containerd.io

[osboxes@localhost ~]$ sudo dnf install docker-ce docker-ce-cli containerd.io
Last metadata expiration check: 0:01:10 ago on Mon 08 Mar 2021 06:48:15 AM EST.
Dependencies resolved.
=====
Package                                Architecture  Version                Repository              Size
=====
Installing:
docker-ce                              x86_64        3:20.10.5-3.fc33       docker-ce-stable        25 M
docker-ce-cli                          x86_64        1:20.10.5-3.fc33       docker-ce-stable        33 M
Installing dependencies:
container-selinux                      noarch        2:2.145.0-1.fc33       fedora                   38 k
containerd.io                          x86_64        1.4.3-3.1.fc33         docker-ce-stable        32 M
docker-ce-rootless-extras              x86_64        20.10.5-3.fc33         docker-ce-stable        5.4 M
fuse-overlayfs                          x86_64        1.4.0-1.fc33           updates                  73 k
libcgroup                              x86_64        0.42.2-3.fc33          fedora                   66 k
slirp4netns                            x86_64        1.1.8-1.fc33           updates                  56 k
=====
Transaction Summary
=====
Install 8 Packages

Total download size: 96 M
Installed size: 419 M
Is this ok [y/N]: y
```

Tijdens de installatie zal gevraagd worden om de GPG key te controleren. Momenteel is dit de officiële fingerprint die je zou moeten zien:

```
060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35
```

Indien je inderdaad de juiste fingerprint krijgt kan je opnieuw bevestigen met **y**.

```
osboxes@fedora:~ — sudo dnf install docker-ce do...

Importing GPG key 0x621E9F35:
  Userid      : "Docker Release (CE rpm) <docker@docker.com>"
  Fingerprint: 060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35
  From        : https://download.docker.com/linux/fedora/gpg
Is this ok [y/N]: y
```

Nadat de installatie voltooid is krijg je opnieuw de controle over de terminal.

## Docker starten en status controleren

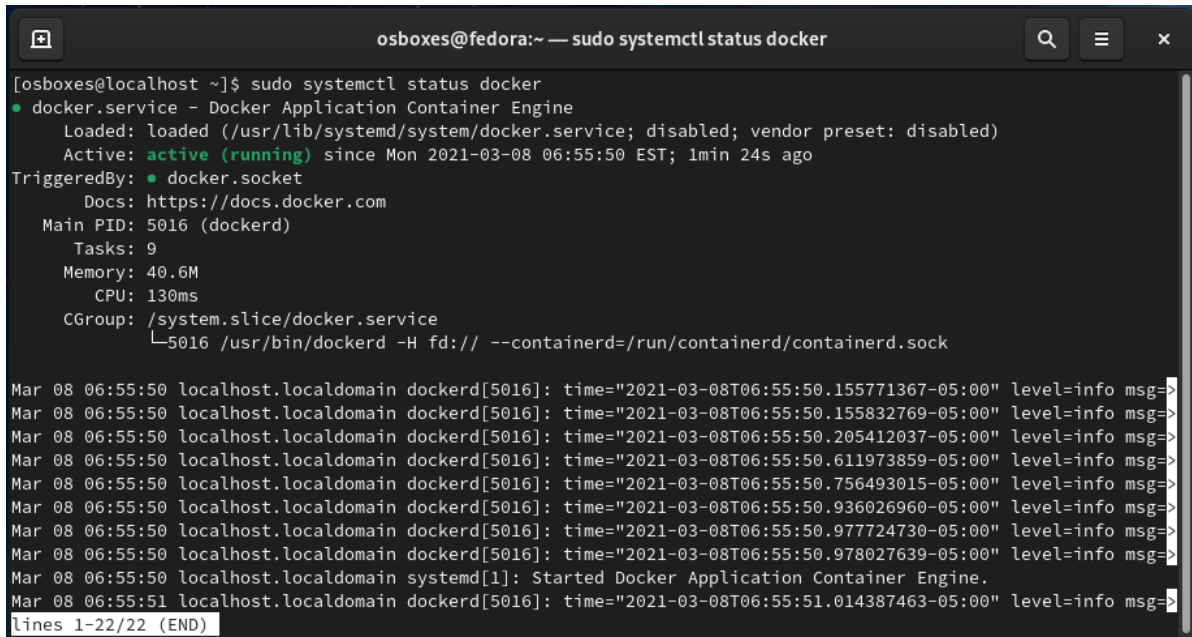
Nadat docker geïnstalleerd is, moeten we deze nog starten. De Docker engine draait als een daemon (service) die we kunnen starten via het *systemctl start* commando:

```
sudo systemctl start docker
```

Na starten van de service, kunnen we de status controleren via het *systemctl status* commando:

```
sudo systemctl status docker
```

Indien alles correct verlopen is, zal je zien dat de service als status *active (running)* heeft:

A terminal window titled 'osboxes@fedora:~ — sudo systemctl status docker'. The output shows the status of the Docker service. It is loaded, active (running), and has been running since Mon 2021-03-08 06:55:50 EST. The service is triggered by docker.socket and has a main PID of 5016 (dockerd). It shows memory usage of 40.6M and CPU usage of 130ms. The CGroup is /system.slice/docker.service. Below this, there is a list of log messages from dockerd and systemd, showing the service starting successfully. The terminal output is as follows:

```
[osboxes@localhost ~]$ sudo systemctl status docker
• docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-03-08 06:55:50 EST; 1min 24s ago
   TriggeredBy: • docker.socket
     Docs: https://docs.docker.com
    Main PID: 5016 (dockerd)
      Tasks: 9
     Memory: 40.6M
        CPU: 130ms
    CGroup: /system.slice/docker.service
            └─5016 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.155771367-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.155832769-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.205412037-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.611973859-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.756493015-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.936026960-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.977724730-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:50.978027639-05:00" level=info msg=>
Mar 08 06:55:50 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
Mar 08 06:55:51 localhost.localdomain dockerd[5016]: time="2021-03-08T06:55:51.014387463-05:00" level=info msg=>
lines 1-22/22 (END)
```

Door `q` in te drukken keer je terug naar de terminal.

## Service persistent maken (*optioneel*)

Bij herstarten van je Linux VM zal de Docker service nog niet automatisch gestart worden. Je kan deze dan wel opnieuw herstarten via het `systemctl start` commando, maar het is ook mogelijk om er voor te zorgen dat de service automatisch gestart wordt bij elke reboot. Indien gewenst, kan je dit instellen via het `systemctl enable` commando:

```
sudo systemctl enable docker
```

## Hello Docker World!

We hebben de Docker engine nu gestart, maar we hebben nog geen container opgestart. Om een container te starten kunnen we zelf een image maken, of we kunnen ook een bestaande image gebruiken, zoals in onderstaand voorbeeld:

```
sudo docker run hello-world
```

Dit commando zal een test image downloaden, en deze nadien uitvoeren in een container. Tijdens de uitvoering schrijft de container een boodschap naar het scherm, waarna de container afsluit.

```
osboxes@fedora:~  
[osboxes@localhost ~]$ sudo docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
b8dfde127a29: Pull complete  
Digest: sha256:89b647c604b2a436fc3aa56ab1ec515c26b085ac0c15b0d105bc475be15738fb  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
[osboxes@localhost ~]$
```

Je kan op elk moment controleren welke containers er momenteel actief zijn op je systeem via het *docker ps* commando:

```
sudo docker ps
```

Welke containers zie je?

Je zal merken dat er wellicht nog geen containers draaien, ook al hebben we net een container opgestart. De reden hiervoor is dat een container niet automatisch blijft draaien, maar afgesloten wordt van zodra alle taken (hier: een tekst printen op het scherm) afgerond zijn. Je kan echter in het vorige commando een extra optie meegeven om ook afgesloten containers te zien:

```
sudo docker ps -a
```

```
osboxes@fedora:~  
[osboxes@localhost ~]$ sudo docker ps  
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES  
[osboxes@localhost ~]$ sudo docker ps -a  
CONTAINER ID   IMAGE     COMMAND   CREATED      STATUS      PORTS      NAMES  
d40664d3a12d   hello-world  "/hello"   7 minutes ago  Exited (0)  7 minutes ago  crazy_meninsky  
[osboxes@localhost ~]$
```

In dit overzicht zie je de unieke ID van elke container, de image die gebruikt wordt (hier *hello-world*), wanneer de container aangemaakt is en wat de huidige status is. Elke container krijgt ook een naam, en indien je zelf geen naam expliciet meegeeft wordt een pseudo-random naam gegenereerd (in bovenstaande screenshot is dit *crazy\_meninsky*).

## Extra containers

Je kan zelf een image maken voor een container, maar Docker beschikt ook over een grote repository van images, Docker Hub (<https://hub.docker.com/>), die iedereen kan gebruiken. Dankzij Docker Hub kunnen we dus snel andere containers opstarten. Bovendien is het zo dat, als je zelf een container wil maken, je meestal vertrekt van een bestaande container en deze dan verder uitbreidt.

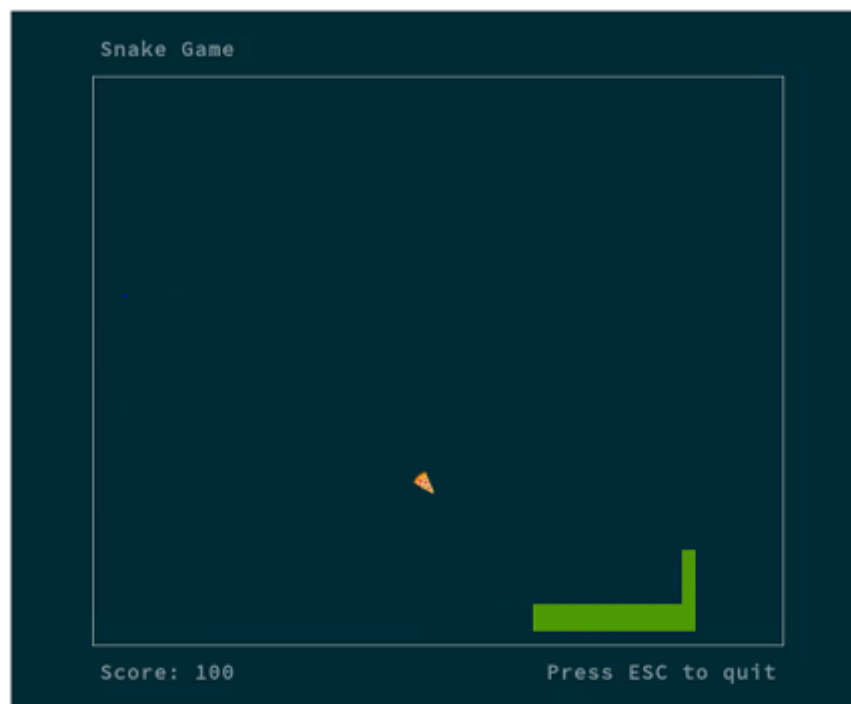
Onderstaand voorbeeld zal een container opstarten op basis van de *chuanwen/cowsay* image. Deze container print een random [fortune](#) boodschap via het [cowsay](#) commando:

```
sudo docker run chuanwen/cowsay
```

```
/ Shapiro, thank you for this useful link\  
\ Be a bad boy and get started to work  /  
-----  
      ^__^  
      (oo)\_____  
      (____)\       )\/\  
              ||----w |  
              ||     ||
```

Een andere image maakt bijvoorbeeld een container waarin je snake kan spelen:

```
sudo docker run -ti dyego/snake-game
```



💡 Zoek zelf eens naar een paar bestaande images op Docker Hub en probeer deze te starten. Noteer de commando's om deze op te starten hieronder:

## Containers en images beheren en verwijderen

Een container wordt binnen Docker dus opgestart via een image. Als je voor de eerste keer een container start, zal de image mogelijk nog gedownload moeten worden, maar als je in de toekomst nieuwe containers opstart die gebaseerd zijn op dezelfde image moet je de image gelukkig niet opnieuw downloaden (tenzij die ondertussen geüpdated zou zijn op de centrale repository).

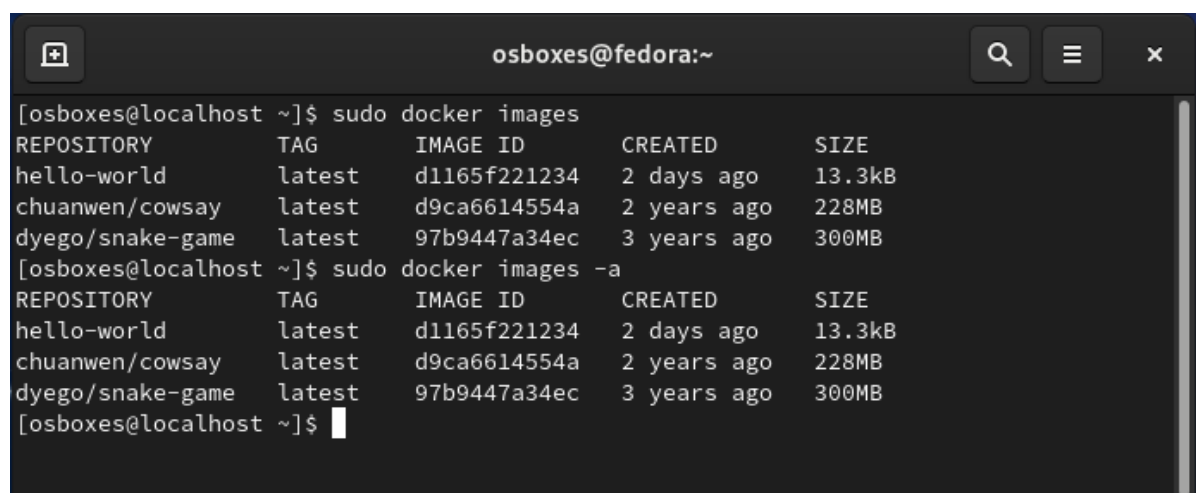
Je kan op elk moment een overzicht opvragen van alle images die je reeds gedownload hebt via het *docker images* commando:

```
sudo docker images
```

Je kan aan bovenstaand commando ook de optie *-a* meegeven om ook intermediate (tussenliggende) images te zien:

```
sudo docker images -a
```

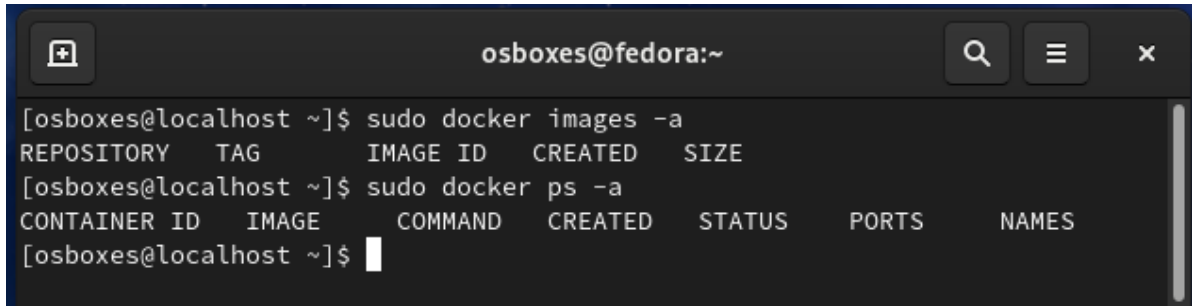
? Is er een verschil in de output van beide commando's? Wat kan je hieruit besluiten?

A terminal window titled 'osboxes@fedora:~' showing the output of 'sudo docker images' and 'sudo docker images -a'. Both commands list three images: 'hello-world', 'chuanwen/cowsay', and 'dyego/snake-game'. The output is identical for both commands.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	d1165f221234	2 days ago	13.3kB
chuanwen/cowsay	latest	d9ca6614554a	2 years ago	228MB
dyego/snake-game	latest	97b9447a34ec	3 years ago	300MB

Tenslotte kan je via het *docker images* commando ook de lokale versies van images verwijderen, en via het *docker container* commando kan je (afgesloten) containers verwijderen. Zoek zelf eens op hoe je op jouw VM **alle containers** en **alle images** kan verwijderen. Noteer de nodige commando's hieronder:

Je kan het resultaat controleren via de commando's *docker images* en *docker ps*:



```
osboxes@fedora:~  
[osboxes@localhost ~]$ sudo docker images -a  
REPOSITORY    TAG       IMAGE ID      CREATED      SIZE  
[osboxes@localhost ~]$ sudo docker ps -a  
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES  
[osboxes@localhost ~]$
```

## Extra informatie

- Docker Engine installeren op Fedora 32/33  
<https://docs.docker.com/engine/install/fedora/>
- Uitgebreide tutorial Docker  
<https://www.youtube.com/watch?v=fqMOX6JJhGo>
- Beheer van docker images  
<https://docs.docker.com/engine/reference/commandline/images/>
- Beheer van docker containers  
<https://docs.docker.com/engine/reference/commandline/container/>