

Systems Advanced Docker Containers

Container Volumes



Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be



Docker container storage

Met docker volumes kunnen we een directory in de container mounten op een directory van de host voor persistente data of niet-persistente, tijdelijke data.

Zonder volumes worden de files in de writable layer geplaatst, die sowieso wordt mee verwijderd met `docker container rm <CONTAINER>`.

Verschillende types van Volumes

- **Anonymous volumes**
 - Een anonymous volume is een volume waarvoor docker een directory op de host zelf toewijst en beheert.
 - `docker run -v /path/in/container ...`
 - wordt samen met de container verwijderd
- **Named volumes**
 - Een named volume is een volume waarvoor docker een directory op de host zelf toewijst, maar jij geeft het volume een naam en beheert het.
 - `docker volume create somevolumename`
 - `docker run -v somevolumename:/path/in/container ...`
 - wordt niet samen met de container verwijderd, moet apart verwijderd worden
- **Host volumes of bind mounts**
 - beste optie voor persistent data
 - Een host volume of bind mount is een volume dat gemount is op een zelf gekozen directory van de host. Jij beheert het volume en de directory op de host.
 - `docker run -v /path/on/host:/path/in/container ...`
 - `docker run -v /path/to/file/on/host:/path/in/container ...`
 - om een specifieke file op de host in een container volume te mounten
 - wordt niet samen met de container verwijderd, moet apart verwijderd worden
- **(tmpfs)**
 - Een volume wordt door docker gemount op een filesystem in het RAM geheugen van de host.
 - wordt samen met de container verwijderd

Anonymous volumes

Een anonymous volume is een volume waarvoor docker een directory op de host zelf toewijst en beheert.

Locatie van anonymous container volumes op de host:

- Op Windows wordt de `/var/lib/docker/` directory gemount op het Hyper-V volume `\\wsl$\\docker-desktop-data\\version-pack-data\\community\\docker\\` en moet je daar gaan zoeken naar de volume directories van docker.
- Op linux kan je gewoon naar de `/var/lib/docker/...` locatie gaan die uit de `docker inspect` komt.

Verwijderen:

`docker container rm -v <CONTAINER>`

- met de `-v` optie verwijder je anonymous volumes samen met de container.

```
# thraa @ DESKTOP-TOMC in ~\docker-les [00:12:03]
$ docker run -it -v /containervol alpine sh
/ # echo "Hallo, PXL!" > /containervol/hallo
/ # exit

# thraa @ DESKTOP-TOMC in ~\docker-les [00:12:34]
$ docker container ls --all
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
2b42cc2ddc47   alpine    "sh"      18 seconds ago   Exited (0) 4 seconds ago           festive_thompson

# thraa @ DESKTOP-TOMC in ~\docker-les [00:12:38]
$ docker inspect festive_thompson | grep -A5 Mounts
"Mounts": [
  {
    "Type": "volume",
    "Name": "ff42d900a0aa20b7bebc3b320cb4b2152a567f0a879cb2e52c2918a3462067fd",
    "Source": "/var/lib/docker/volumes/ff42d900a0aa20b7bebc3b320cb4b2152a567f0a879cb2e52c2918a3462067fd/_data",
    "Destination": "/containervol",

# thraa @ DESKTOP-TOMC in ~\docker-les [00:13:03]
$ ls \\wsl$\\docker-desktop-data\\version-pack-data\\community\\docker\\volumes\\ff42d900a0aa20b7bebc3b320cb4b2152a567f0a879cb2e52c2918a3462067fd\\_data
hallo

# thraa @ DESKTOP-TOMC in ~\docker-les [00:14:15]
$ cat \\wsl$\\docker-desktop-data\\version-pack-data\\community\\docker\\volumes\\ff42d900a0aa20b7bebc3b320cb4b2152a567f0a879cb2e52c2918a3462067fd\\_data\\hallo
Hallo, PXL!

# thraa @ DESKTOP-TOMC in ~\docker-les [00:14:29]
$ docker container rm -v festive_thompson
festive_thompson

# thraa @ DESKTOP-TOMC in ~\docker-les [00:15:39]
$ cat \\wsl$\\docker-desktop-data\\version-pack-data\\community\\docker\\volumes\\ff42d900a0aa20b7bebc3b320cb4b2152a567f0a879cb2e52c2918a3462067fd\\_data\\hallo
Get-Content: Cannot find path '\\wsl$\\docker-desktop-data\\version-pack-data\\community\\docker\\volumes\\ff42d900a0aa20b7bebc3b320cb4b2152a567f0a879cb2e52c2918a3462067fd\\_data\\hallo' because it does not exist.

# thraa @ DESKTOP-TOMC in ~\docker-les [00:15:45] C:1
$ _
```

Named volumes

Een named volume is een volume waarvoor docker een directory op de host zelf toewijst, maar jij geeft het volume een naam en beheert het.

Je kan het op voorhand aanmaken, maar dat hoeft niet

docker volume create <VOLUME>

Een named volume kan gedeeld worden onder verschillende containers, zolang je het mount.

**docker run -v <VOLUME>:</path/in/container>
<CONTAINER>**

Verwijderen moet manueel gebeuren. Wanneer je de container verwijderd met -v, zal dat geen effect hebben op het volume.

Manueel verwijderen kan met **docker volume rm <VOLUME>**

```
# thraa @ DESKTOP-TOMC in ~\docker-les [00:53:49]
$ docker volume rm mijnvolume
mijnvolume

# thraa @ DESKTOP-TOMC in ~\docker-les [00:53:54]
$ _
```

```
# thraa @ DESKTOP-TOMC in ~\docker-les [00:38:34]
$ docker volume create mijnvolume
mijnvolume

# thraa @ DESKTOP-TOMC in ~\docker-les [00:39:11]
$ docker volume ls
DRIVER      VOLUME NAME
local      mijnvolume

# thraa @ DESKTOP-TOMC in ~\docker-les [00:39:17]
$ docker volume inspect mijnvolume
[
  {
    "CreatedAt": "2022-10-20T22:39:11Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/mijnvolume/_data",
    "Name": "mijnvolume",
    "Options": {},
    "Scope": "local"
  }
]

# thraa @ DESKTOP-TOMC in ~\docker-les [00:39:28]
$ docker run -it -v mijnvolume:/containervol alpine sh
/ # echo "Hallo, PXL!" > /containervol/hallo
/ # exit

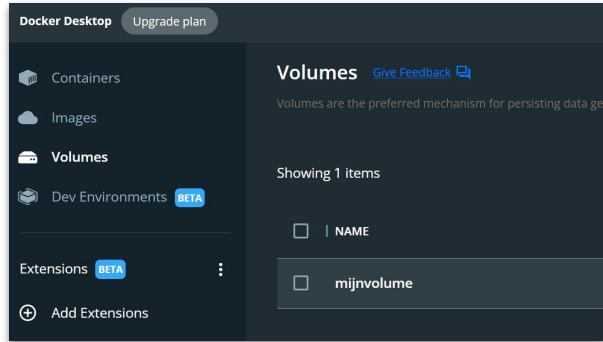
# thraa @ DESKTOP-TOMC in ~\docker-les [00:40:11]
$ docker inspect festive_johnson | grep -A5 Mounts
"Mounts": [
  {
    "Type": "volume",
    "Name": "mijnvolume",
    "Source": "/var/lib/docker/volumes/mijnvolume/_data",
    "Destination": "/containervol",

# thraa @ DESKTOP-TOMC in ~\docker-les [00:40:39]
$ cat \\wsl$\docker-desktop-data\version-pack-data\community\docker\volumes\mijnvolume\_data\hallo
Hallo, PXL!

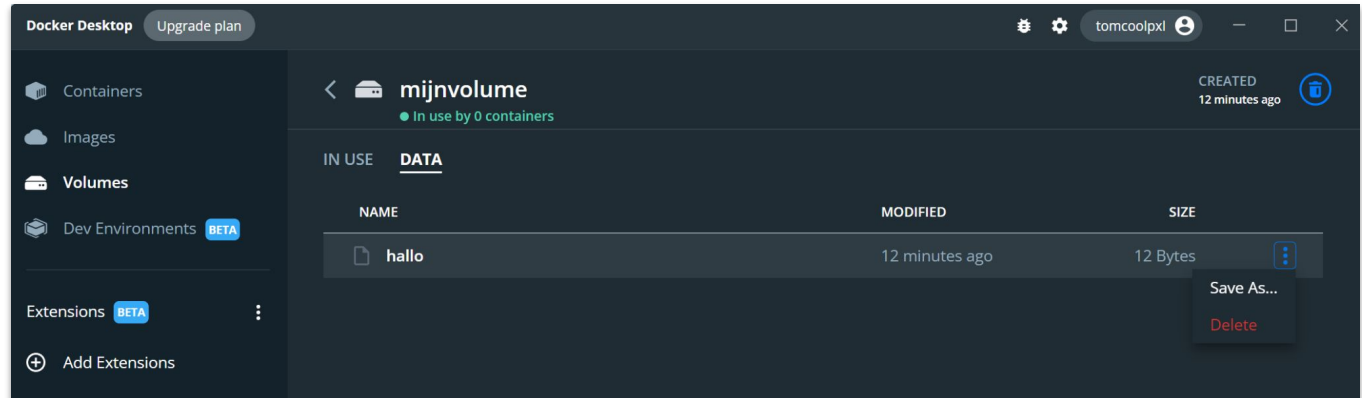
# thraa @ DESKTOP-TOMC in ~\docker-les [00:41:01]
$ docker container rm -v festive_johnson
festive_johnson

# thraa @ DESKTOP-TOMC in ~\docker-les [00:41:17]
$ cat \\wsl$\docker-desktop-data\version-pack-data\community\docker\volumes\mijnvolume\_data\hallo
Hallo, PXL!
```

Named volumes en Docker Desktop



Docker Desktop geeft een snel overzicht van volumes en hun gebruik, maar is verder eerder beperkt.



Host volumes

Een **host volume** of **bind mount** is een volume dat gemount is op een bestaande directory van de host. Jij beheert zowel het volume als de directory op de host.

```
docker run -v  
/path/on/host:/path/in/container <CONTAINER>
```

Je kan ook een specifieke file op de host in een container volume te mounten met

```
docker run -v  
/path/to/file/on/host:/path/in/container  
<CONTAINER>
```

Verwijderen moet manueel gebeuren. Wanneer je de container verwijderd met -v, zal dat geen effect hebben op het volume.

Manueel verwijderen kan je gewoon door de directory te verwijderen.

```
# thraa @ DESKTOP-TOMC in ~\docker-les [00:58:35]
$ mkdir hostvol

Directory: C:\Users\thraa\docker-les

Mode                LastWriteTime         Length Name
----                -
d----             21/10/2022           0:58     hostvol

# thraa @ DESKTOP-TOMC in ~\docker-les [00:58:40]
$ echo "mijn tekst" > hostvol/mijnfile

# thraa @ DESKTOP-TOMC in ~\docker-les [00:59:05]
$ cat hostvol/mijnfile
mijn tekst

# thraa @ DESKTOP-TOMC in ~\docker-les [00:59:19]
$ docker run -it -v C:\Users\thraa\docker-les\hostvol:/containervol/ alpine sh
/ # cat /containervol/mijnfile
mijn tekst
/ # echo "tweede regel" >> /containervol/mijnfile
/ # exit

# thraa @ DESKTOP-TOMC in ~\docker-les [01:00:48]
$ cat hostvol/mijnfile
mijn tekst
tweede regel

# thraa @ DESKTOP-TOMC in ~\docker-les [01:00:55]
$ docker container rm -v modest_buck
modest_buck

# thraa @ DESKTOP-TOMC in ~\docker-les [01:01:26]
$ cat hostvol/mijnfile
mijn tekst
tweede regel

# thraa @ DESKTOP-TOMC in ~\docker-les [01:01:30]
$
```

Volumes kunnen gedeeld worden onder containers

We kunnen meerdere containers eenzelfde volume laten gebruiken.

```
docker run -it -v /tmp/vancontainer:/test-volume --name=voltainer1 ubuntu /bin/bash  
echo "hello world" > /test-volume/myfile
```

CTRL-P-Q

```
docker run -it -v /tmp/vancontainer:/test-volume --name=voltainer2 ubuntu /bin/bash  
cat /test-volume/myfile  
hello world
```

CTRL-P-Q

- we speciëren hier ook de directory op de host
- Zo kunnen we meerdere containers hun volume naar dezelfde directory op de host laten wijzen om bestanden ertussen uit te wisselen. (bv voor logs met een gecentraliseerde log-parser)

```
docker run -it --volumes-from=voltainer2 --name=voltainer3 ubuntu /bin/bash  
cat /test-volume/myfile  
hello world
```

- andere manier om volume van een andere container te delen

Volumes mounten met --mount

Je kan in plaats van `--volume (-v)` ook `--mount` (met meer opties) gebruiken

Volume aanmaken

```
docker volume create myvol
```

Volume gebruiken met de parameter `-v` of `--volume`

```
docker run -it -v myvol:/app --name=volumetest ubuntu /bin/bash
```

Volume gebruiken met de parameter `--mount`

```
docker run -it --mount source=myvol,target=/app --name=mounttest ubuntu /bin/bash
```

Voor meer info:

<https://docs.docker.com/storage/bind-mounts/>

Volume data verwijderen

Hoe kan je de data van een container weggooien

Container zonder volumes

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

data zit in een writable layer van de container

Container met anonymous volume

```
docker container rm -v <container>
```

data zit in een 'ongekende' directory op de host

Container met named volume

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

```
docker volume rm <volumename>
```

data zit in een 'ongekende' directory op de host,
maar is benaderbaar via een zelfgekozen naam

Container met host volume

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

```
rm -rf <hostdir>
```

data zit in een 'zelfgemaakte' directory op de host,
en deze directory kan verwijderd worden via de shell

OEFFENING

Werken met volumes

- Maak op de Dockerhost de directory **/dockerfiles** aan
 - plaats in deze directory een file
 - genaamd **testfile**
 - met de tekst "Dit is een test"
- Start een Alpine container die de inhoud van deze file toont
- Pas de tekst van **testfile** aan op de host
 - naar "Dit is een tweede test"
- Herstart de container om de nieuwe inhoud te zien
- Start opnieuw een Alpine container die de inhoud van deze file toont
- Start een nieuwe Alpine container waarmee je via de shell de inhoud van de file kan wijzigen
 - ga naar een prompt binnen deze container om de inhoud van de file te tonen
 - verander vanuit de container de tekst van de file naar "Aangepast vanuit de Alpine container"
- Controleer op de host of de file veranderd is
- Start een nieuwe Ubuntu container waarmee je via de shell de inhoud van de file kan wijzigen
 - ga naar een prompt binnen deze container om de inhoud van de file te tonen
 - verander vanuit de container de tekst van de file naar "Aangepast vanuit de Ubuntu container"
- Controleer op de host of de file veranderd is

