doc

Het ip adres van de docker host is te vinden via de docker quickstart terminal in "C:\Program Files\Docker Toolbox"

(start.sh)

Opstarten in deze terminal duurt best een poos, de default docker vm wordt hier opgestart.

Omdat we met docker toolbox werken (geen pro versie van windows 10) moet een ander ip adres gebruikt worden ipv localhost

Vanuit deze terminal geeft het volgende commando het ip adres:

docker-machine ip

De laatste keer was dit (en dat zal wel zo blijven)

192.168.99.100

De docker voorbeelden uit bijvoorbeeld https://docs.docker.com/get-started/part2/ werken via dit ip adres,

zo zal het python voorbeeld dat wordt opgestart via docker run -p 4000:80 friendlyhello

te vinden zijn op het adres vanuit de browser:

http://192.168.99.100:4000/

start postgres docker image op in een container met

docker run --name mypostgres --net host -e POSTGRES\_PASSWORD=mysecretpassword -d -p 5432:5432 postgres

Via het commando

docker container ls

is het mogelijk om te achterhalen wat het id is van dit docker proces laten we zeggen

309a6a850721

Start nu van binnen deze container de postgres sql op:

docker exec -it 309a6a850721 bash

Vanuit de bash prompt de postgres sql starten als gebruiker postgres

psql -U postgres

Dan connecten met de juiste database

\c mytest

Hier kun je de tabel infotable bekijken:

select \* from infotable;

Database aanmaken

create database mytest;

Table aanmaken

create table infotable(naam varchar(30));

insert into infotable(naam) values ('adriaan');

commit hoeft niet, autocommit.

Vanuit tooltje TablePlus een connectie maken, name een goede naam (postgres default db) Tag (development)

host het machine ip adres van docker - in plaats van localhost (192.168.99.100), port 5432, user postgres en

password mysecretpassword.

het is bijna 19:00 ik ga eten!

Als xxxx het korte id is van de container (te zien bij docker container ls):

docker container stop xxxx

kan de container weer opnieuw worden opgestart met behoud van data

probeer eerst

docker container start xxxx

Dit geeft een foutmelding

docker: Error response from daemon: Conflict. The container name "/mypostgres01" is already in use by

container "309a6a850721161447165d9ec0f0da3e2b9f0d1e36d83ff9592a6d9ec57577f9".

Deze lange id (laten we zeggen xxxxxxxxxxxxx) gebruiken om de container opnieuw op te starten:

docker container start xxxxxxxxxxxxx

De data is dan nog steeds aanwezig. Op dezelfde manier kun je een container opnieuw opstarten nadat de pc opnieuw is

opgestart. Ook dan geldt dat de data er dan nog steeds is.

Na een container remove van de postgres container is de data verdwenen. Bedoeling is om hier een volume van

te maken dat bewaard blijft nadat de container is verwijderd.

Ubuntu gebruiken

docker run --name ubuntu\_bash --rm -i -t ubuntu bash

Volume maken

docker volume create my-vol

--rm

Volume inspecteren

docker volume ls

Volume verwijderen

docker volume rm my-vol

|  |
| --- |
| ik werd hier helemaal niet lekker van: Ik kwam er niet goed achter hoe “live” een lokale folder kon worden gebruikt om met een docker container bestanden uit te wisselen: wat betekent onderstaande nu allemaal?? |
| Start container met een volume  docker run -d \  --name devtest \  --mount source=myvol2,target=/app \  nginx:latest |
| dus heb ik maar met een voorbeeld gedaan waarin een dockerfile werd gebruikt om een image te maken. Dockerfile: |
| # Use an official Ubuntu runtime as a parent image  FROM ubuntu:latest  # Set the working directory to /app  WORKDIR /app  # Copy the current directory contents into the container at /app  COPY . /app |
| docker image build -t ubuntu:1.0 .  en dan  docker container run --name ubuntu --rm -i -t ubuntu:1.0 bash |

Creating a shared folder and accessing it from container

<https://stackoverflow.com/questions/35315996/how-do-i-mount-a-docker-volume-while-using-a-windows-host>

cmd window.

Stop alle containers.

Stop de virtual machine (VBoxManage bevindt zich in de Oracle VirtualBox folder: C:\Program Files\Oracle\VirtualBox).

vboxmanage controlvm default poweroff

om het aanmaken van een shared folder in VirtualBox mogelijk te maken

VBoxManage sharedfolder add default --name "dockerdata" --hostpath D:\dockerdata

Start de docker vm via de docker quickstart terminal (docker toolbox!)

check of de docker vm machine draait:

docker-machine.exe active

zou moeten tonen: default (de default vm machine).

docker-machine.exe ssh default 'sudo mount -t vboxsf dockerdata /d/dockerdata'

## Creating shared folder op host Windows is niet gelukt. Andere manier:

maak een special purpose ubuntu image met enig doel om de bestanden te kopiëren naar een gemeenschappelijk volume genaamd my-vol

|  |
| --- |
| FROM ubuntu:latest  VOLUME d:/dockerdata  RUN mkdir /app  WORKDIR /app  COPY . /app |

Bouw hiervan een image

|  |
| --- |
| docker image build -t ubuntu:1.0 . |

(Let op de afsluitende punt, nl de huidige folder van waaruit dit commando wordt gestart).

Run dit image op een manier dat het volume wordt aangekoppeld. We kopiëren de bestanden naar /media

|  |
| --- |
| container run --name ubuntu --rm --mount source=my-vol,target=/media -i -t ubuntu:1.0 bash |

Het mount point moet in een lege folder zijn. In dit geval /media. Het volume my-vol moet al zijn aangemaakt via het commando *docker volume create my-vol*

Door het bouwproces zijn de bestanden naar /app gekopiëerd vanuit de huidige folder

We kunnen nu vanuit /app de bestanden naar het permanentere volume my-vol kopiëren: bijvoorbeeld het bestand hello.TXT dat zich in de werkfolder bevond:

|  |
| --- |
| cp hello.TXT media |

exit uit de container. De ubuntu container en het ubuntu image kunnen nu zelfs worden verwijderd via docker image rm en docker container rm.

Draai nu een andere container en connect deze met het volume:

|  |
| --- |
| docker container run --name alpine --rm --mount source=my-vol,target=/media -i -t alpine sh |

In /media zal nu het bestand hello.TXT te vinden zijn.

Uiteraard wordt dit bestand niet bijgewerkt wanneer op de windows host de inhoud wordt gewijzigd. De container maakt gebruik van een virtual disk.

Dit is nu éénrichting verkeer, het is vast wel mogelijk om een FTP server als container te bouwen en via FTP de bestanden van en naar de host te sturen.

Inlezen tsv file

Ik ben er niet achter gekomen hoe je bij een tsv een header kunt negeren dus header eraf via een editor of via sed:

sed 1d title\_basics.tsv > title\_basics01.tsv

vervolgens inlezen, kan even duren bij een groot bestand

|  |
| --- |
| copy actor(nconst, primary\_name, birth\_year, death\_year, primary\_profession, known\_for\_titles)  from '/media/name\_basics01.tsv'  DELIMITER E'\t'; |