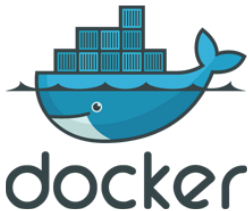


Docker

Kan ik er iets nuttig mee doen ?

Nikolas Garofil

18 mei 2018



- Zoveel als gewenst isoleren
- Hetgeen in de isolatie zo klein maken als gewenst
- Contact met de buitenwereld zo laag houden als gewenst

- Zoveel als gewenst isoleren
- Hetgeen in de isolatie zo klein maken als gewenst
- Contact met de buitenwereld zo laag houden als gewenst

Onverwachte features zoveel mogelijk gebruiken.

Hoe het er vaag uitziet

Neem het beste van:

- virtuele machines:
- chroot-containers:

Wat mengen en nog wat extra's bijgooien.

Hoe het er vaag uitziet

Neem het beste van:

- virtuele machines: De eenvoudigheid, de isolatie
- chroot-containers:

Wat mengen en nog wat extra's bijgooien.

Hoe het er vaag uitziet

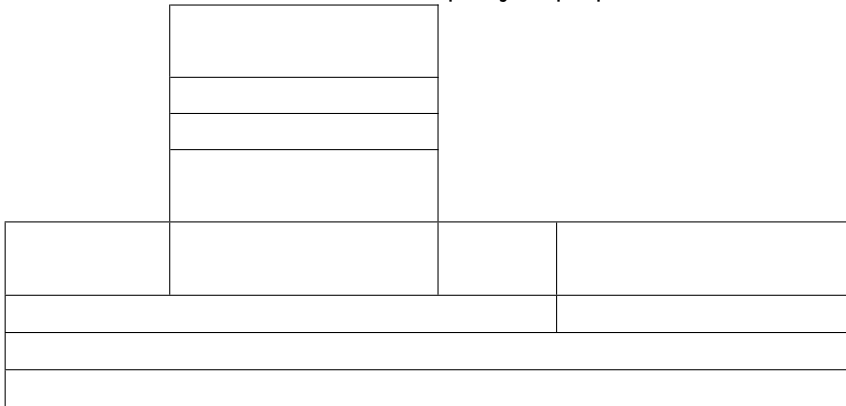
Neem het beste van:

- virtuele machines: De eenvoudigheid, de isolatie
- chroot-containers: Het “gewicht”, de nabijheid bij de host

Wat mengen en nog wat extra's bijgooien.

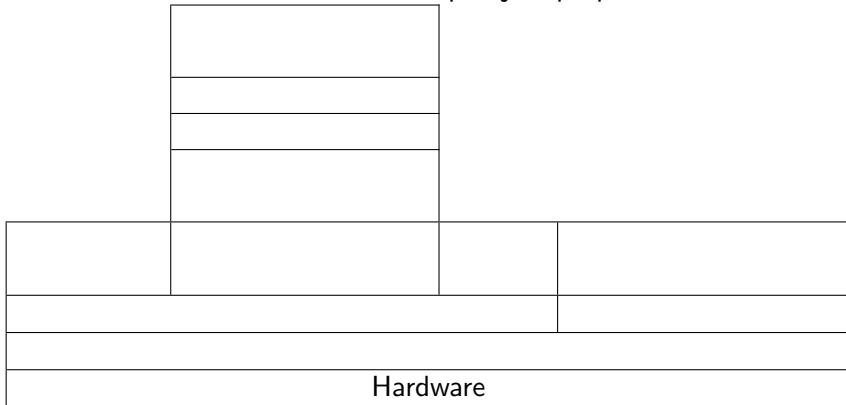
Docker iets technischer:

De software stack op mijn laptop:



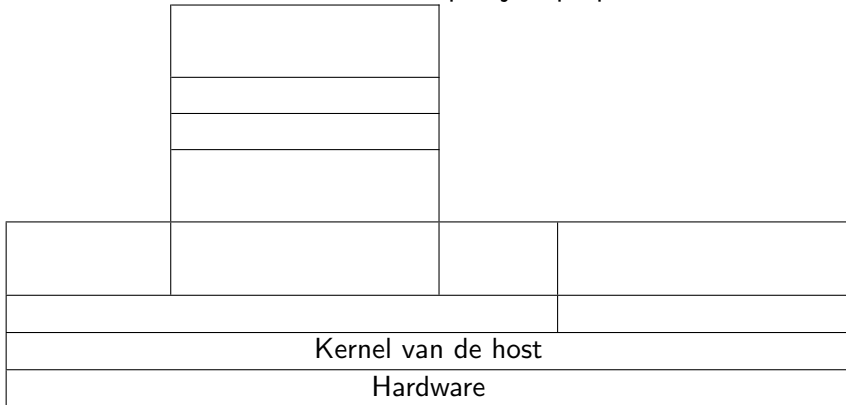
Docker iets technischer:

De software stack op mijn laptop:



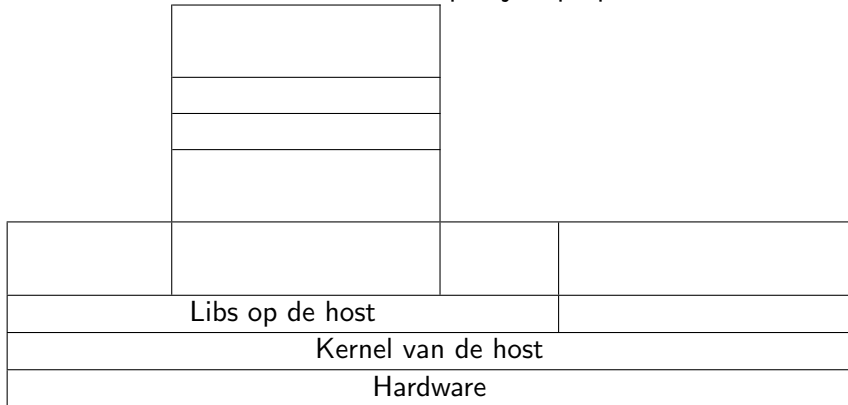
Docker iets technischer:

De software stack op mijn laptop:



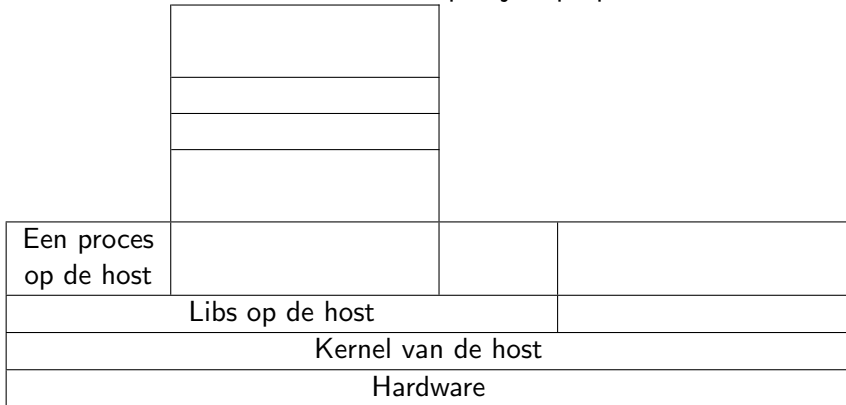
Docker iets technischer:

De software stack op mijn laptop:



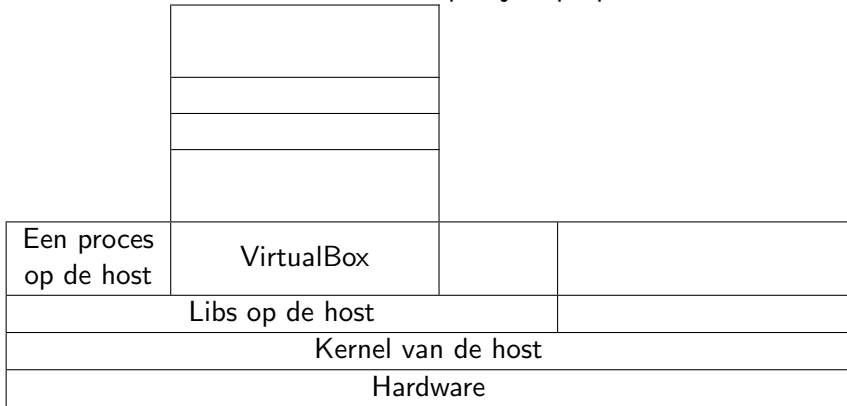
Docker iets technischer:

De software stack op mijn laptop:



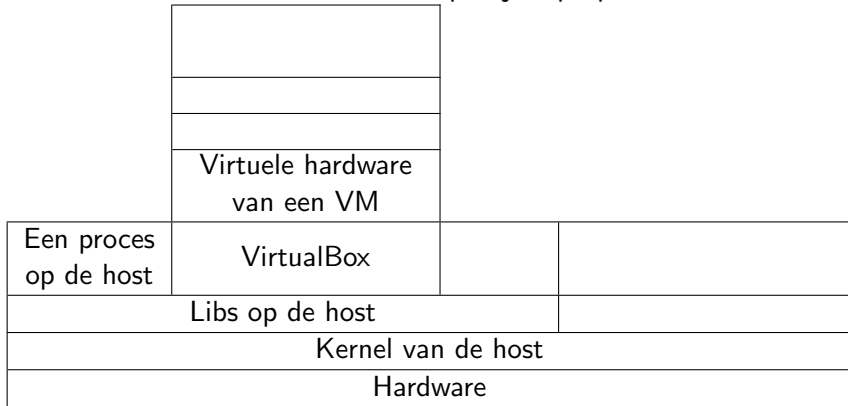
Docker iets technischer:

De software stack op mijn laptop:



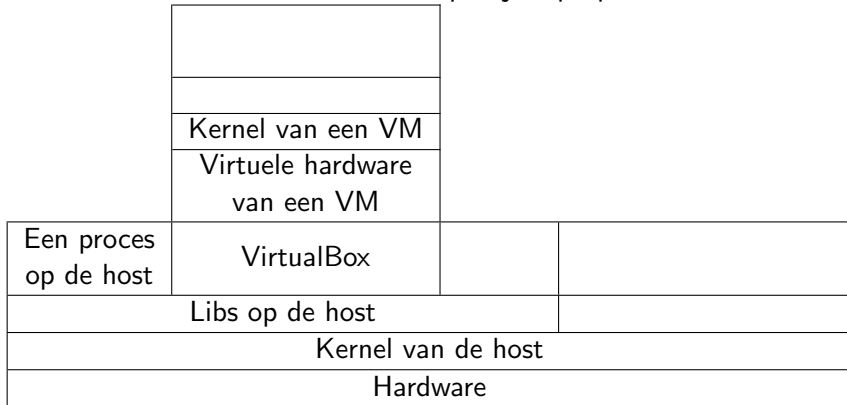
Docker iets technischer:

De software stack op mijn laptop:



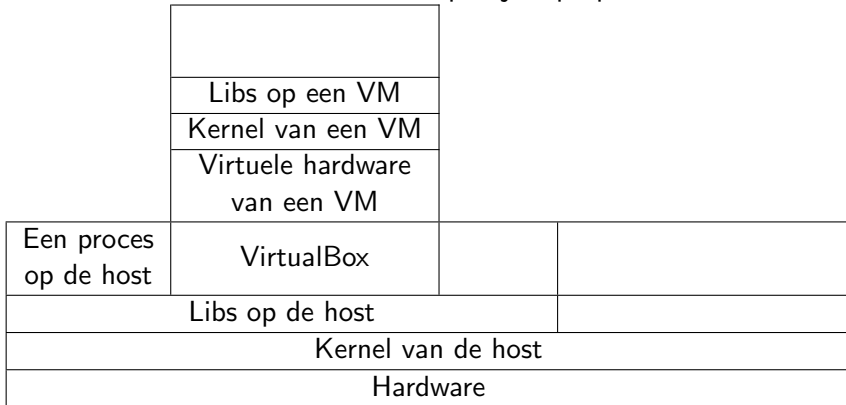
Docker iets technischer:

De software stack op mijn laptop:



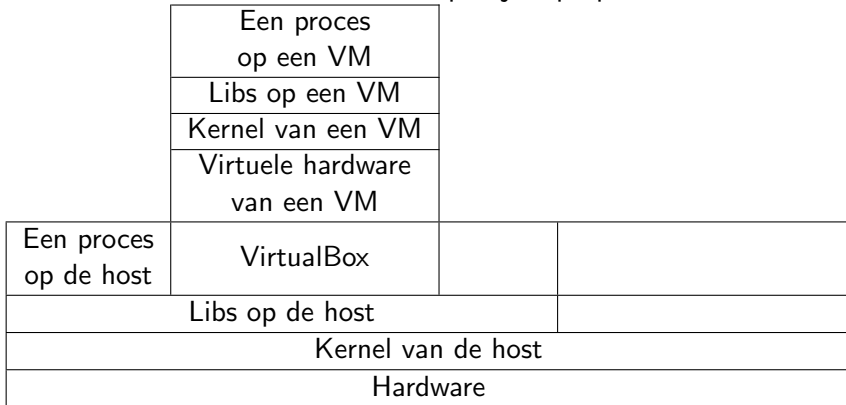
Docker iets technischer:

De software stack op mijn laptop:



Docker iets technischer:

De software stack op mijn laptop:



Docker iets technischer:

De software stack op mijn laptop:

	Een proces op een VM		
	Libs op een VM		
	Kernel van een VM		
	Virtuele hardware van een VM		
Een proces op de host	VirtualBox	Docker	
Libs op de host			
Kernel van de host			
Hardware			

Docker iets technischer:

De software stack op mijn laptop:

	Een proces op een VM		
	Libs op een VM		
	Kernel van een VM		
	Virtuele hardware van een VM		
Een proces op de host	VirtualBox	Docker	
Libs op de host			Libs op een container
Kernel van de host			
Hardware			

Docker iets technischer:

De software stack op mijn laptop:

	Een proces op een VM		
	Libs op een VM		
	Kernel van een VM		
	Virtuele hardware van een VM		
Een proces op de host	VirtualBox	Docker	Een proces op een container
Libs op de host			Libs op een container
Kernel van de host			
Hardware			

Onmiddelijk zichtbare voordelen:

- Isolatie keuzemogelijkheden
 - Netwerken, TCP/UDP poorten
 - Geheugen, Filesystem en Devices
- Volledig alternatieve distro's in containers
- Minimale containers voor exact 1 service

Onmiddelijk zichtbare voordelen:

- Isolatie keuzemogelijkheden
 - Netwerken, TCP/UDP poorten
 - Geheugen, Filesystem en Devices
- Volledig alternatieve distro's in containers
- Minimale containers voor exact 1 service
- Resource management
 - Rekenkracht
 - Geheugen
 - Schijf

- Vergelijkbaar met een snapshot van een schijf
- Basisimages voor elke bekende distro en Alpine

- Vergelijkbaar met een snapshot van een schijf
- Basisimages voor elke bekende distro en Alpine
- Copy-On-Write lagen
 - Door commits (gelijkaardig aan version control systems)
 - Op basis van een Dockerfile

- Verplaatsbaarheid zowel images als containers
- Verder bouwen met minder werk
- DockerHub en GitHub koppeling
- Overzichtelijkheid (bij Dockerfile-images)

1 Begin-image

- 1 Begin-image
- 2 Definities hoe elke laag moet opgebouwd worden:
 - Commando's
 - Bestanden van host kopiëren
 - Specialekes: Environment var's, omgeving aanpassingen, ...

- 1 Begin-image
- 2 Definities hoe elke laag moet opgebouwd worden:
 - Commando's
 - Bestanden van host kopiëren
 - Specialekes: Environment var's, omgeving aanpassingen, ...
- 3 Standaard PID 1

- Grote systeem aanpassingen met laag risico
- Identieke containers tegelijk draaien
- Snel uittesten onbekende software
- Eenvoudig uittesten onbekende software
- Clean-room testen
- Reproduceerbaarheid fouten
- ...

Cheatsheet meest gebruikte commandos

- `docker build -t foo /ergens`
Image foo bouwen als Dockerfile in /ergens zit.
- `docker run OPTIES foo`
foo downloaden, met OPTIES container maken/starten
- `docker ps -a ; docker inspect bar`
Korte info alle containers; alle info over container bar
- `docker stop|start|pause|unpause bar`
Container bar zijn toestand aanpassen
- `docker exec [-it] bar /path/naar/commando`
Extra commando [interactief] uitvoeren in bar
- `docker logs bar`
Alle uitvoer (STDOUT en STDERR) van de container tonen.

Cheatsheet meest gebruikte opties voor docker run

- `--name bar` : Naam bar geven aan de container
- `-d` of `-ti` : Als daemon of interactief starten
- `-rm` : Container wissen na het stoppen
- `-v /extern:/intern:ro`
Het /extern path wordt in de container read-only beschikbaar als intern. Zonder :ro is het read-write.
- `-p 12:34` of `-P`
TCP/12 op host binden aan TCP/34 in container. Of alle vrijgegeven poorten koppelen aan random host-poorten.
- Varianten van `-e DISPLAY -v`
`$HOME/.Xauthority:/root/.Xauthority --device /dev/snd -v /dev/shm:/dev/shm`
GUI-apps met geluid draaien. Hangt sterk af van de image, X-server config op de host en het geluidssysteem op de host.

Eenvoudige Dockerfile

- FROM `debian:latest`
Image maken op basis van de laatste debian image
- RUN `apt-get update && apt-get -y install vim`
Vim installeren
- COPY `vimrc /etc/vimrc`
Een vim-configfile vanuit de dir met Dockerfile gebruiken
- CMD `vim`
vim uitvoeren als een container gestart wordt en er geen commando is opgegeven

- `docker-ce` van dockersite halen, niet van uw distro's repos
- Bij maken van images voor de wereld:
Eerst een github rep aanmaken, dockerhub later regelen
- Soms effe `docker image prune` om rommel op te ruimen
- Te kopiëren bestanden in de dir met Dockerfile zetten
- Bij zoeken op Dockerhub, images in deze volgorde zoeken:
"Official" > "Automized build" > Leesbare "Dockerfile" >
Leesbare readme > Veel sterren/downloads.