

Navigatie

Toon huidige directory	pwd
Toont bestanden	ls
Toont bestanden in detail	ls -l
Toon verborgen bestanden	ls -a
Toont bestands groottes	ls -lh
Verander naar een directory	cd
Ga een directory hoger	cd ..
Ga naar home directory	cd
Ga naar rood directory	cd /

Bestandsbeheer

Maak directory	mkdir
Maak onderliggende directories	mkdir -p </oefening/test/>
Maak directory met permissies	mkdir -m <775> <map>
Maak bestand	touch
Kopieer directory	cp
Kopieer incl subdirectories	cp -r
Verplaats / hernoem	mv
Verwijder	rm
Forceer verwijderen	rm -f
Verwijder incl subniveau's	rm -r
Wijzig eigenaar van bestand	chown <user>:<group> <bestand>

Netwerk

Toon netwerkconfiguratie	ifconfig
Test verbinding met een host	ping <host>
Toon info netwerkinterfaces	iwconfig
Scan voor open poorten	nmap <host>
Toon luisterende poorten	netstat -tulnp

Systeeminfo

Toon schrijfruimte	df -h
Toon RAM gebruik	free -h
Toon systeeminfo	uname -a
Toon info systeembronnen	top
Toon actieve processen	ps aux
Toon gebruikersnaam	whoami

Overige

Handleiding commando	man <command>
Redirect output naar bestand	<command> > <bestand>
Append output aan bestand	<inhoud> >> <bestand>

influxDB

image	influxdb:2.7
data volume	/var/lib/influxdb2
config volume	/etc/influxdb2
setup environment	- DOCKER_INFLUXDB_INIT_MODE=setup
gebruiker environment	- DOCKER_INFLUXDB_INIT_USERNAME=<gebruiker>
password environment	- DOCKER_INFLUXDB_INIT_PASSWORD=<wachtwoord>
organisatie environment	- DOCKER_INFLUXDB_INIT_ORG=<org naam>
bucket environment	- DOCKER_INFLUXDB_INIT_BUCKET=<naam bucket>
token maken	docker compose exec <naam .yaml voor influxdb> influx auth list > token (containers moeten up zijn)

Nginx

image	nginx
ports	"80:80"
html volume	/usr/share/nginx/html
config volume	/etc/nginx/nginx.conf:ro
htpasswd volume	/etc/nginx/htpasswd



Grafana	
image	grafana/grafana-oss:9.5.9-ubuntu
data volume	/var/lib/grafana
publieke dashboards environment	- GF_FEATURE_TOGGLES_ENABLE=publicDashboards

Container Manueel / image bouwen	
container manueel starten	docker run -p <port>:<port> --name <naam kiezen> -d <image>
container automatisch weg na gebruik	docker run <zoals hierboven> --rm <image>
mount bestand aan een image file	docker run <zoals hierboven> -v <./-map:image pad> <image>
toon manueel gestarte containers	docker ps -a
verwijder manueel gestarte containers	docker rm <container>
container stoppen	docker stop <container>
root shell starten in container	docker exec -it <container> /bin/bash
container opslaan als image	docker commit <container> <image naam kiezen>
image verwijderen	docker rmi <image:latest>

Image Naar Dockerhub	
repo maken op dockerhub	inloggen > repositories > create repository > geef naam > create
inloggen op docker	docker login
one-time code invoeren	ga naar url > code invoeren > confirm
maak image	zie "image bouwen" (bv test-image)
tag maken voor test-image	docker tag <test-image:latest> <kaitraupel/test-image:latest>
image naar dockerhub pushen	docker push <kaitraupel/test-image>

Mosquitto	
image	eclipse-mosquitto
ports	"1883:1883"
config volume	/mosquitto/config/mosquitto.conf
mosquitto.conf poort	port 1883
.conf anonieme verbinding	allow_anonymous true
.conf user + ww	<passwd> -c /etc/mosquitto/passwd <gebruiker>
.conf log events naar bestand	log_dest file /var/log/mosquitto/mosquitto.log (er onder) log_type all
.conf berichten opslaan	persistence true (er onder) persistence_file mosquitto.db
.conf max verbindingen instellen	max_connections 100
.conf max berichtgrootte	message_size_limit 1024
.conf QoS instellen	qos <0,1 of 2>
clients installeren	sudo apt install mosquitto-clients
subscribe op topic	mosquitto_sub -h localhost -t <topic>
Publish bericht	mosquitto_pub -h localhost -t <topic> -m <"bericht">

Node-Red	
image	nodered/node-red
ports	"8080:1880"
data volume	/data



Docker

Toon actieve container	<code>docker ps</code>
Toon alle container (ook gestopte)	<code>docker ps -a</code>
Toon beschikbare images	<code>docker images</code>
Toon enkel image IDs	<code>docker images -q</code>
Start specifieke container	<code>docker run <image></code>
Start container op achtergrond	<code>docker run -d</code>
Start container op een poort	<code>docker run -p <port:port></code>
Geef container een naam	<code>--name</code>
Command uitvoeren in actieve container	<code>docker exec -it <container> <command></code>
Stop container	<code>docker stop <container></code>
Verwijder container	<code>docker rm <container></code>
Bouw image vanuit Dockerfile	<code>docker build -t <image></code>
Mount directory in container	<code>-v <directory>:<directory></code>
Forceer verwijderen container	<code>docker rm -f <container></code>
Verwijder container incl volumes	<code>docker rm -v <container></code>
Verwijder volumes	<code>docker compose down --volumes</code>
Herbouw image	<code>docker compose up --build</code>
Bekijk logs	<code>docker compose logs</code>
Volg logs in real-time	<code>docker compose logs --follow</code>
Image halen uit Docker Hub	<code>docker pull <image></code>
Commands uitvoeren zonder sudo	<code>sudo usermod -aG docker \$USER</code> (erna uitloggen)
Toon gebruikers ID + groep	<code>id</code>

Docker (cont)

Verander eigenaar van map	<code>sudo chown <ID>:<groep> <map></code>
gebruiker en ww instellen	<code>htpasswd -c htpasswd <gebruiker></code>
htpasswd installeren	<code>sudo apt install apache2-utils</code>
bestand uitvoerbaar maken	<code>chmod u+x <bestand></code>
bestand uitvoeren	<code>sudo ./<bestand></code>
token aanmaken	<code>docker compose exec <naam .yml voor influxdb> influx auth list > token</code> (containers moeten up zijn)
Toon alle container (up en down)	<code>docker ps -a</code>
Toon geïnstalleerde images	<code>docker image ls</code>

Github

Github installeren	<code>sudo apt install git</code>
repo van github clonen	<code>git clone <url></code> (in home repo doen!)

Python In Container Uitvoeren

python image installeren	<code>docker pull python</code>
python map maken	<code>mkdir <map></code> (bv: python)
vorige map koppelen aan container	<code>docker run --rm -it -v ./python:/<bv py> <python image> /bin/bash</code>
uitloggen	<code>exit</code>
script in python map maken	<code>vim python/<naam script></code> (bv test.py)
terug container starten	<code>docker run --rm -it -v ./python:/<bv py> <python image> /bin/bash</code>
naar py map gaan	<code>cd <gekozen mapnaam></code> (bv py/)
kijken of script erin staat	<code>ls</code>
script testen	<code>python <naam script></code> (bv test.py)



Python In Container Uitvoeren (cont)

script extern in	<code>docker run --rm -it -v ./python:/py <image</code>
container uitvoeren	<code>python> python <pad bv /py/test.py></code>

Dockerfiles (python)

Dockerfile maken	<code>vim Dockerfile (let op hoofdletter)</code>
------------------	--



By **KaiTraupel**

cheatography.com/kaitraupel/

Not published yet.

Last updated 13th January, 2025.

Page 4 of 4.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>