

Verslag

Verbinden met SSH

Ga naar settings van je VM en voeg een netwerkadaptor toe, klik op die "Add" knop en dan de optie "Network Adapter"

Ga naar die nieuwe network adapter en kies als network connection "host-only"

Start nu de VM op en voer het commando `ip a` in: dit geeft een extra network interface met state DOWN met de naam `ens37`

Doe nu het commande "`sudo nano /etc/netplan/50-cloud-init.yaml`"

Voeg nu onder version: 2 met dezelfde uitlijning als het eerste netwerk "`ens37: en dhcp4: true`" toe. Doe dan `ctrl x` en opslaan

Als dit gedaan is doe het commando "`sudo netplan try`" en dan bevestig you met enter

Doe nog eens `ip a` en noteer het adress: 192.168.157.129

Verbinden met de SSH doen we in putty of powershell (`ssh <user>@<host>`)

SSH inloggen met keys

Doe het commando `ssh-keygen -t rsa` in windows powershell

Geef de volgende in voor de key "`C:\Users\<username>\.ssh\id_rsa`" ik heb gekozen voor `id_rsa`

GEEN PASSPHRASE

Voer de volgende command uit: `type C:\Users\tobia\.ssh\id_rsa.pub | ssh tobias@192.168.157.129 "cat >> ~/.ssh/authorized_keys"`

Geef dan je wachtwoord in, en als alles goed gaat, is je key gekopieerd.

Log opnieuw in op je VM met de volgende command: `ssh -i C:\Users\tobia\.ssh\ubuntu_server_rsa tobias@192.168.157.129`

Config file

Als alles in de vorige stappen zijn gelukt kan je inloggen zonder wachtwoord op je VM via je host os. Maar je moet elke keer het IP-adres van je VM invullen en de locatie van je key meegeven en dat is vervelend. We gaan dat nu configureren zodat het zelfs nog gemakkelijker gaat.

- Ga naar je de `.ssh` folder (`cd C:\Users\<username>\.ssh`)
- Als je de inhoud van deze folder bekijkt, vind je hier je keys terug, maar waar nu geïnteresseerd in zijn, is de file genaamd `config`.

- Controleer even of deze bestaat met de volgende command: dir (dit is de ls equivalente in windows)
- Als de file niet bestaat kan je deze aanmaken met de volgende command: type nul > config (dit is de ls equivalente in windows)
- Open deze file met de notepad editor: notepad config
 - Voeg hier de volgende config:

Host ubuntu-server

IdentityFile C:\Users\<username>\.ssh\ubuntu_server_rsa

HostName <host>

User <user>

- Om je wijzigingen op te slaan druk je op Control + S
- Log nu terug in op je VM maar met de volgende command: ssh ubuntu-server
- En als je als juist hebt gedaan zou je nu ingelogd moeten zijn zonder je wachtwoord, IP-adres, username of path van de key in te vullen.

Docker

Installeer docker op je VM

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64]

https://download.docker.com/linux/ubuntu focal stable"

apt-cache policy docker-ce

sudo apt install docker-ce

sudo systemctl start docker

Pull nu de docker image door sudo docker pull httpd in te geven

Ik heb een directory gemaakt genaamd HTTPD door mkdir HTTPD te doen

Ik heb in die map een Dockerfile gemaakt met het touch commando

Ik heb ook een directory gemaakt genaamd public-html

In de dockerfile heb ik FROM httpd:2.4

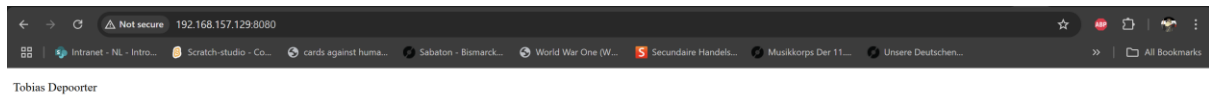
COPY ./public-html/ /usr/local/apache2/htdocs/

Gekopieerd

In de public-html heb ik een html file gemaakt met het touch commando genaamd index.html en daarin heb ik de html code gestoken

Daarna heb ik docker build -t my-apache2 . getypt dit built de docker image

Vervolgens heb ik docker run -dit --name my-running-app -p 8080:80 my-apache2 gedaan voor dit te runnen



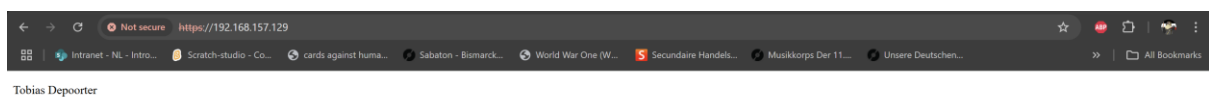
Dit is wat ik bij de dockerfile heb gezet om de https juist te krijgen. Ik heb het mv commando moeten gebruiken om example.key/crt naar server.key.crt te krijgen

```
FROM httpd:2.4
COPY ./public-html/ /usr/local/apache2/htdocs/

COPY server.crt /usr/local/apache2/conf
COPY server.key /usr/local/apache2/conf

RUN sed -i \
    -e 's/^#(Include .*httpd-ssl.conf)/\1/' \
    -e 's/^#(LoadModule .*mod_ssl.so)/\1/' \
    -e 's/^#(LoadModule .*mod_socache_shmcb.so)/\1/' \
    conf/httpd.conf
```

En dit is het commando om het te runnen: `sudo docker run -dit --name my-running-file -p 443:443 my-apache2`



Dit is de Docker images foto (ik snap niet goed wat die 3 months ago zijn want dit is een nieuwe VM)

```
tobias@ubuntu-tobias:~/HTTPD$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-apache2	latest	8b9de50ffc3f	4 minutes ago	148MB
<none>	<none>	f8aa15816bba	16 minutes ago	148MB
<none>	<none>	62a81747dda0	35 minutes ago	148MB
<none>	<none>	a3e8a684ee57	35 minutes ago	148MB
<none>	<none>	e3b1b3e135f9	2 hours ago	148MB
hello	latest	085cc59a303c	2 hours ago	174MB
httpd	2.4	1bcf11fa154f	3 months ago	148MB
httpd	latest	1bcf11fa154f	3 months ago	148MB

```
tobias@ubuntu-tobias:~/HTTPD$
```

Dit is de Docker ps commando

```
tobias@ubuntu-tobias:~/HTTPD$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
04984b8db6f0	my-apache2	"httpd-foreground"	5 minutes ago	Up 5 minutes	80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp	my-running-file
aa2bc6f9e923	a3e8a684ee57	"httpd-foreground"	23 minutes ago	Up 23 minutes	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	my-running-apps

```
tobias@ubuntu-tobias:~/HTTPD$
```