

> Start nieuwe regel

regel met spatie na afbreking

regel zonder spatie na afbreking

Nested Virtualization met WSL2

WSL is superhandig, veelzijdig en volop in ontwikkeling. Er zijn twee smaken WSL; WSL-1 (die System Calls bedoelt voor de Linux Kernel, vertaald naar WinNT Kernel System Calls - eigenlijk WINE, maar dan andersom) en WSL-2 die gebruikt maakt van een zogenaamde Lightweight Utility VM. > **Henk den Hartog**

De Lightweight Utility VM biedt veel mogelijkheden en een leuke hiervan is wel Nested Virtualisatie, oftewel het kunnen draaien van Virtuele Machines in een Virtuele Machine (de Lightweight Utility VM in het geval van WSL-2).

Dit is één van de nieuwe features die wordt geïntroduceerd in de aankomende update van WSL2, maar met wat kunst en vliegwerk kun je hier ook al mee spelen op basis van de huidige versie van WSL2. In dit artikel ga je KVM installeren op een Ubuntu WSL-2 distributie. De werkwijze is getest op hardware met een Intel processor (support voor AMD volgt pas later) met de volgende uitgaven van Windows10:

- > Windows Professional 10.0.19042 N/A Build 19042
- > Windows Enterprise 10.0.19042 N/A Build 19042
- > Windows Home 10.0.19042 Build 19042

Let wel de installatie van WSL-2 valt buiten de scope van dit artikel, maar gelukkig kun je een duidelijke how to vinden op de site van Microsoft:

<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

THIRD PARTY VIRTUALISATIE SOFTWARE

Voordat je begint, is het goed je te realiseren dat WSL-2 gebruikmaakt van virtualisatie en dus Hyper-V zal activeren, zij het een uitgeklede versie. Dit betekent dat virtuele machines gemaakt met third party virtualisatie software als VMWare Player en VirtualBox niet meer zullen werken, zolang de WSL-2 feature aanstaat. Microsoft biedt een additionele feature aan, genaamd "Windows Hypervisor Platform" die een API voor dit doeleinde aanbiedt.

```

root@DESKTOP-AV1PCDT: /mnt/c/Users/henk
PS C:\Users\henk> wsl -d kvm
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.72-microsoft-standard-WSL2: x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat May  8 14:08:33 CEST 2021

System load:  0.0          Processes:    8
Usage of /:   6.2% of 250.98GB  Users logged in:  0
Memory usage: 0%          IPv4 address for eth0: 172.23.224.6
Swap usage:   0%

0 updates can be installed immediately.
0 of these updates are security updates.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@DESKTOP-AV1PCDT:/mnt/c/Users/henk# kvm-ok
INFO: Your CPU does not support KVM extensions
KVM acceleration can NOT be used
root@DESKTOP-AV1PCDT:/mnt/c/Users/henk#

```

^ WSL-2 heeft out of the box geen virtualisatie actief staan.

De laatste versie van VirtualBox kan hiermee omgaan, maar zit nog in de experimentele fase en lijkt in de praktijk niet goed te werken.

EEN EXTRA DISTRIBUTIE INSTALLEREN

Je begint met het creëren van een extra WSL-distributie. Je wilt voor dit experiment namelijk niet je eventuele huidige WSL Ubuntu installatie aanpassen, maar je kunt deze daarvoor wel gebruiken als je wil. Nog een keer dezelfde distributie halen uit de Microsoft Store gaat niet, daarom gebruiken we Powershell om een extra instantie van

Ubuntu20.04 te downloaden. Start powershell en maak in je homedirectory een wsl-distros directory:

- ```

> cd ~
> mkdir wsl-distros
> cd wsl-distros

```

Download nu Ubuntu met de Invoke-WebRequest cmdlet:

- ```

> Invoke-WebRequest -Uri https://cloud-images.ubuntu.com/focal/current/focal-server-cloudimg-amd64-wsl.rootfs.tar.gz -OutFile kvm.tar.gz -UseBasicParsing

```

Nb1. In plaats van gebruik te maken van Invoke-WebRequest

kun je natuurlijk ook handmatig een image downloaden van <https://cloud-images.ubuntu.com>.

Nb2. Dit duurt een tijdje; in de tussentijd kun je alvast verder gaan met het installeren van WinDbg Preview uit de Microsoft Store en het downloaden van de benodigde scripts (verderop in dit artikel beschreven)

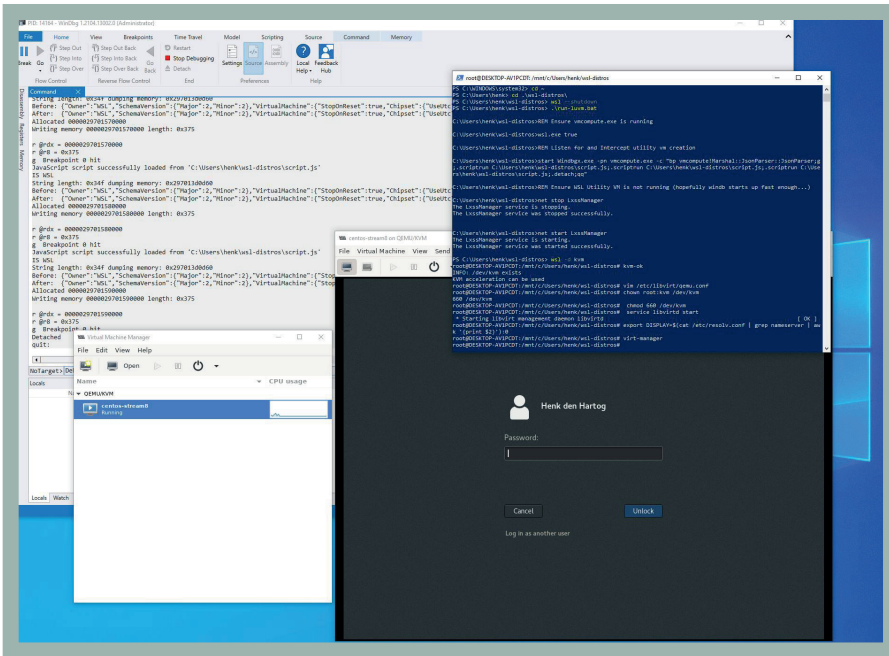
Importeer de Ubuntu WSL distributie onder de naam "kvm" met wsl import:

- ```

> wsl --import kvm ./kvm kvm.tar.gz --version 2

```

nb. Let op dat je importeert als



Centos8 Stream VM op KVM met WSL-2 op Windows 10.

version 2. Je kunt nu je nieuwe “kvm” WSL-distributie starten:

```
> wsl -d kvm
```

Update en installeer vervolgens de volgende pakketten:

```
> apt update
> apt upgrade
> apt -y install build-essential
libncurses-dev bison flex
libssl-dev libelf-dev cpu-
checker qemu-kvm libvirt-daemon-
system libvirt-clients bridge-
utils libvirt-daemon-system-sysv
virt-manager
```

Wanneer je nu het commando ‘kvm-ok’ uitvoert, krijg je het volgende te zien:

```
INFO: Your CPU does not support
KVM extensions
KVM acceleration can NOT be used
```

Dat ga je nu fixen door het geschikt maken van de Lightweight Utility VM voor Nested Virtualisatie.

## NESTED VIRTUALISATIE

Nu ga je de Lightweight Utility VM geschikt maken voor Nested Virtualisatie. Voor een normale Hyper-V VM kun je dit aanzetten met de “Set-VM-Processor” Powershell CMDlet, maar de Lightweight Utility VM wordt op een andere manier gemanaged (met HCS en daarom vind je deze bijvoorbeeld ook niet terug in de Hyper-V manager).

Bovendien wordt de Lightweight Utility VM elke keer opnieuw aangemaakt bij het starten van de eerste

WSL-instantie en dus moet je op een andere manier nested virtualisatie activeren.

Dit ga je doen met behulp van WinDbg Preview en een aantal scripts welke te vinden zijn op de volgende GitLab pagina: <https://gitlab.com/conclusionxforce/automation/public/wsl2-nested-virtualization>

nb. Deze scripts zijn afgeleid van het origineel, welke te vinden zijn op: <https://gist.github.com/steffengy/62a0b5baa124830a4b0fe4334ccc2606>.

Ga naar de Microsoft Store, zoek naar “WinDbg Preview” en installeer deze. Stop vervolgens de Lightweight Utility VM, maar let op! Dit impliceert ook dat al je reeds draaiende WSL-distributies de nek worden omgedraaid. Sluit deze dus eerst netjes af!

Start Powershell en typ in:

```
> wsl --shutdown
```

Haal het run-luv.m.bat en script. js van de Conclusion Xforce git repository en plaats deze ergens op je systeem. Waar maakt niet zoveel uit, als je ze maar weer terug kunt vinden en ze beide bij elkaar in dezelfde map staan. Start nu Powershell weer, maar dit keer met “Run as Administrator”, want we gaan de LxssManager Windows Services herstarten in het run-luv.m.bat script en dat is een beschermd proces.

Start het run-luv.m.bat script in deze Powershell met Administrator rechten:

```
> ./run-luv.m.bat
```

Wacht tot de Windows Debugger volledig is gestart door dit script. Soms kan het echter voorkomen, met name bij de eerste keer, dat de foutmelding “The RPC server is unavailable” tevoorschijn komt. Dit komt omdat de WinDBG dan niet snel genoeg geïnitieerd is. Let dus goed op deze melding, want dat betekent dat het niet werkt. De oplossing is echter simpel: probeer het nog een keer, nadat je er zeker van bent dat je Powershell als Administrator hebt uitgevoerd en je een `wsl --shutdown` hebt gedaan.

## STARTEN WSL KVM DISTRO

Nu kun je de eerder gecreëerde “kvm” WSL-distributie weer starten:

```
> wsl -d kvm
```

Dit hoeft overigens niet in hetzelfde Powershell venster met Administrator rechten (liever niet zelfs). Wanneer je nu “kvm-ok” opstart, zie je:

```
INFO: /dev/kvm exists
KVM acceleration can be used
```

Dit betekent dat virtualisatie mogelijk is, maar er moet nog een aantal kleine dingen gedaan worden voordat we KVM kunnen starten. Qemu logt standaard naar systemd-logd, maar een WSL-distributie maakt geen gebruik van Systemd. Voeg daarom het volgende toe aan /etc/libvirt/qemu.conf:

```
> stdio_handler = "file"
```

Er moeten wat rechten goed gezet worden op /dev/kvm

```
> chown root:kvm /dev/kvm
> chmod 660 /dev/kvm
```

Nu kun je libvirt starten:

```
> service libvirt start
```

## VIRTUELE MACHINE

Wanneer je op Windows een X server hebt geïnstalleerd, kun je virt-manager starten en een VM creëren. Hoe je een X Server moet installeren onder Windows, valt ook buiten scope van dit artikel, maar in het Linux Magazine nummer 6 van 2019 kun je lezen wat hiervoor nodig is op basis van WSL-1 en VcXsrv. Als je de stappen in dit artikel volgt, loont het de moeite om in de Microsoft Store op zoek te gaan naar het programma GWSL. Ook dit is een gratis X Server voor Windows, maar die is meer geïntegreerd. Zo kun je bijvoorbeeld je grafische applicaties vanuit GWSL starten en D-bus configureren op je distro. Ook moet je je DISPLAY anders instellen voor WSL-2. Voor een WSL-2 distributie draait de X server namelijk niet op localhost maar op een extern IP-adres. Vanuit de WSL-2 distributie gezien is dit het IP-adres van je default gateway. Op deze manier stel je dit IP-adres in als waarde voor de DISPLAY variabele: `> export DISPLAY=$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}'):0`

Wist je trouwens dat in de aankomende grote update van Windows 10 er standaard met WSL-2 een X server geïntegreerd gaat worden?

## TOT SLOT

In dit artikel heb je alvast een kijkje in de toekomst genomen als het gaat om Nested Virtualisatie met WSL-2. WSL-2 en zijn Lightweight Utility VM, die in de volgende release out-of-the-box Nested-Virtualisatie zal ondersteunen, biedt eindeloze mogelijkheden, zeker als je vervolgens KVM installeert. Wat dacht je bijvoorbeeld van MacOS draaien op Windows10 met WSL-2? Deze link vond ik, maar heb ik zelf niet getest:

<https://dev.to/nicole/running-macos-on-windows-10-with-wsl2-kvm-and-qemu-21e1>

Krijg jij het aan de gang? <