* Docker manueel installeren of verwerken in installatiebestand voor hosts
* Host configureren voor telnet
* Via telnet verbinden met host (script)
* Docker aansturen (script)
  + Run docker
  + Dockerfile instructions
  + Build image
  + Run container
  + Verify installation
  + Test package

Werkt op W10 Home:

Run in a **command prompt** as **administrator**:

1. Install Hyper-V

pushd "%~dp0"

dir /b %SystemRoot%\servicing\Packages\\*Hyper-V\*.mum >hyper-v.txt

for /f %%i in ('findstr /i . hyper-v.txt 2^>nul') do dism /online /norestart /add-package:"%SystemRoot%\servicing\Packages\%%i"

del hyper-v.txt

Dism /online /enable-feature /featurename:Microsoft-Hyper-V -All /LimitAccess /ALL

pause

1. Install Containers

pushd "%~dp0"

dir /b %SystemRoot%\servicing\Packages\\*containers\*.mum >containers.txt

for /f %%i in ('findstr /i . containers.txt 2^>nul') do dism /online /norestart /add-package:"%SystemRoot%\servicing\Packages\%%i"

del containers.txt

Dism /online /enable-feature /featurename:Containers -All /LimitAccess /ALL

pause

1. Edit registry keys

REG ADD "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /f /v EditionID /t REG\_SZ /d "Professional"

REG ADD "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /f /v ProductName /t REG\_SZ /d "Windows 10 Pro"

1. **Download and run official Docker Installer For Windows.**

In my case the registry keys were restored after restart, but you could restore them manually:

REG ADD "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v EditionID /t REG\_SZ /d "Core"

REG ADD "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v ProductName /t REG\_SZ /d "Windows 10 Home"

Alternatief: Chocolatey

<https://medium.com/@remisharoon/the-smartest-way-to-run-docker-on-windows-10-home-441c4dd215d>

Alternatief met Linux Subsystem for Windows:

## (1) Setup a Linux instance via WSL2

Get into an elevated CLI prompt (CMD and PS both work fine):

1. Confirm you've got WSL2: wsl --status
2. Install Ubuntu: wsl --install
   * Same as wsl --install --distribution ubuntu
   * Missing component error: wsl –-set-default-version 1
3. After you're forced to reboot, and you log back in, you'll get a specialized command prompt window that pops up. Setup you Linux user and password. (If you forget, you can do a wsl --user root followed by passwd {your-account-name} to fix it.)
4. Update: sudo apt update && sudo apt upgrade
5. Mine didn't have ifconfig: sudo apt install -y net-tools

If you want more info on how to control your instance(s), look at [the Microsoft docs](https://docs.microsoft.com/en-us/windows/wsl/wsl-config).

In general, getting "into" the default, Ubuntu instance within WSL is as easy as typing either "bash" or "ubuntu" from a regular CLI prompt. Though, I'd highly recommend installing "Windows Terminal" and using that instead.

## (2) Install Docker

Open up a prompt inside your Linux instance. The [general instructions](https://docs.docker.com/engine/install/ubuntu/) are here, if you need more help. I used a Ubuntu instance.

1. Trust the docker repo's GPG key: curl -fsSL https://download.docker.com/linux/${ID}/gpg | sudo apt-key add -
   * Same idea as: curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
2. Add their stable repo as a package source: echo "deb [arch=$(dpkg --print-architecture)] https://download.docker.com/linux/${ID} $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
   * Same idea as: echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
3. Update your local index: sudo apt update
4. Install docker!: sudo apt install -y docker-ce docker-ce-cli containerd.io
5. Add your account to the "docker" group: sudo usermod -aG docker $USER
6. Get that group change to be recognized:
   * Close all of your open sessions and wait ~10 seconds for the Linux instance to get shutdown automatically. Then open up a new session.
   * Close the window and force the instance to restart (from a Windows CLI prompt): wsl --terminate {distro-name} . Then open up a new session.
7. Start docker: sudo -b dockerd
8. Prove it's working: docker run --rm hello-world

### Dockerd and services/auto-start concerns

WSL2 doesn't presently have a clean way to auto-start the dockerd daemon. There's a lot of workarounds on the 'Net. Some people start it via a Scheduled task that starts dockerd via wsl. Some people start it via a smart chunk of code in either .profile or .bashrc. Soon, there's supposed to be a [officially supported approach](https://docs.microsoft.com/en-us/windows/wsl/wsl-config) via the "[boot]" section of the /etc/wsl.conf file, but it still doesn't appear to have landed in Windows 10 20H2 as [it was suggested it would](https://www.howtogeek.com/714485/whats-new-in-windows-10s-21h2-update-coming-winter-2021/).

**UPDATE 2021-11-25**: I decided on adding this to my .profile ([copied from here](https://dev.to/felipecrs/simply-run-docker-on-wsl2-3o8)). It goes around sudo.

if service docker status 2>&1 | grep -q "is not running"; then

wsl.exe -d "${WSL\_DISTRO\_NAME}" -u root -e /usr/sbin/service docker start > /dev/null 2>&1

fi

## Interactions with the Windows host

* From a Linux instance into Windows, there's mount points: /mnt/{windows-drive-letter}/...
* From Windows into a Linux instance, there's a magic "share": \\wsl$\{linux-instance-name}\...

Installeren: <https://www.geeksforgeeks.org/how-to-install-linux-packages-inside-a-docker-container/>

https://askubuntu.com/questions/1297571/docker-ubuntu-how-to-permanently-install-software