CUPS Druckerserver

https://hub.docker.com/r/anujdatar/cups

https://github.com/anujdatar/cups-docker

```
services:
    cups:
        image: anujdatar/cups:25.10.01
        container_name: cups
        network_mode: host #optional, falls etwas mit discovern der Geraete nicht klappt
        restart: unless-stopped
        ports:
            - "631:631"
        devices:
            - /dev/bus/usb:/dev/bus/usb
        environment:
            - CUPSADMIN=admin
            - CUPSPASSWORD=MEINPW
            - TZ=Europe/Berlin
        volumes:
            - app:/etc/cups
            #- eigene-treiber:/opt/eigene-treiber
            #- spool:/var/spool/cups
        # Folgendes CMD nur wenn man eigene Treiber als .deb im Volume eingepflegt hat:
        #command: sh -c "dpkg -i /opt/eigene-treiber/cnijfilter2_6.00-1_amd64.deb && apt-get
install -f -y && /entrypoint.sh"

volumes:
  app:
  #eigene-treiber:
  #spool:
```

Falls Port 631 vom System already in use

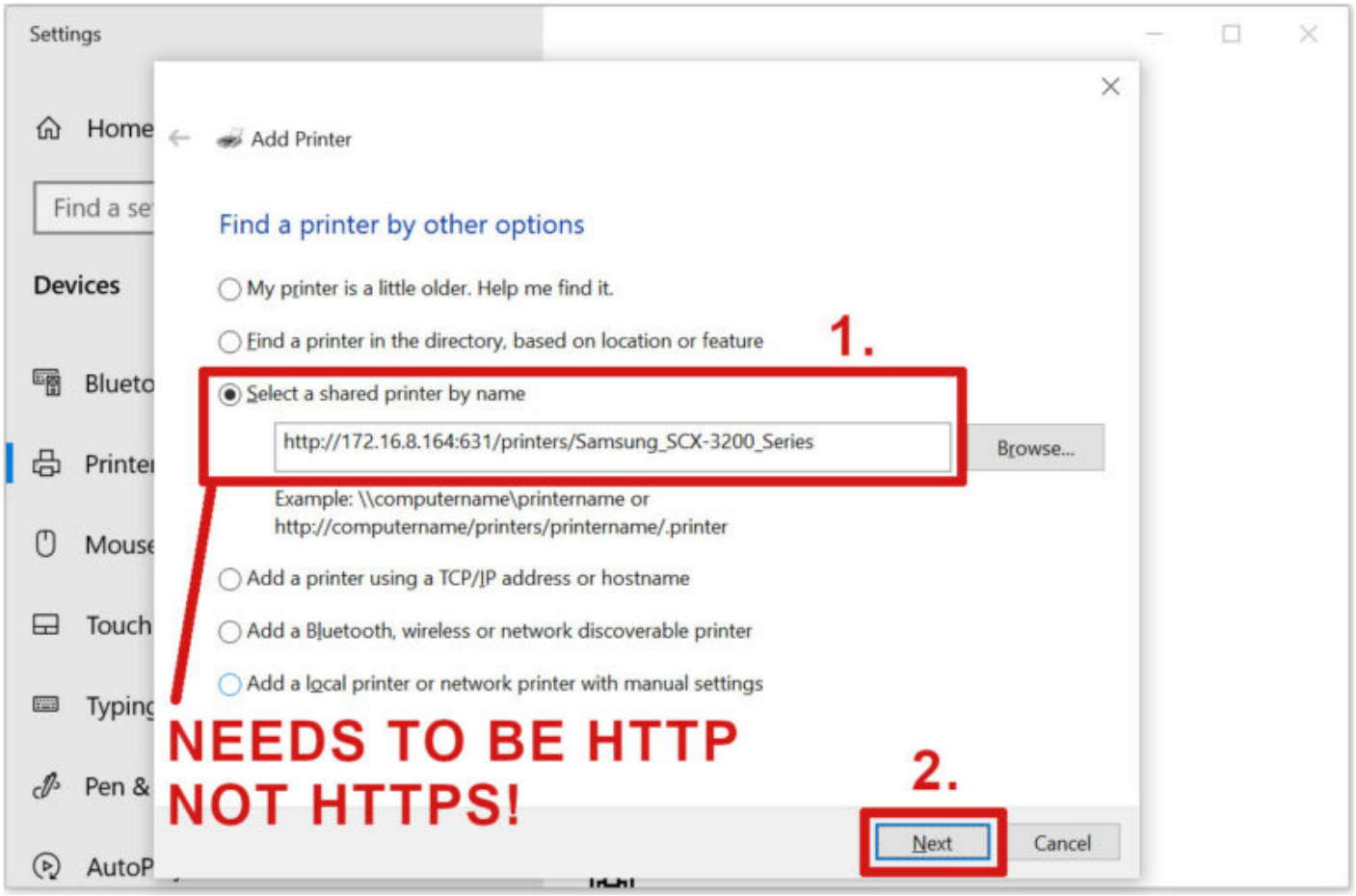
```
sudo systemctl stop cups
sudo systemctl disable cups

sudo apt remove cups

#optional
sudo systemctl stop cups-browsed
sudo systemctl disable cups-browsed
```

Mit CUPS verbundenen Drucker in Windows einfügen

https://techblog.paalijarvi.fi/2020/05/25/making-windows-10-to-print-to-a-cups-printer-over-the-network/



pnputil verwenden um Treiber zu installieren:

https://www.reddit.com/r/PowerShell/comments/14k0wkb/addprinter_ugh/

für ansible aus https://stackoverflow.com/questions/69673238/adding-a-printer-with-powershell:

```
- name: install Sharp MX3070N driver
  win_shell: pnputil /add-driver "C:\sharp-mx3070n\su0emenu.inf" /install

- name: add printer port
  win_shell: Add-PrinterPort -Name "printer3" -PrinterHostAddress "yourprintersIP"

- name: add printer driver
  win_shell: Add-PrinterDriver -Name "SHARP MX-3070N PCL6" # if this string is not known, grab
from .inf file

- name: add printer
  win_shell: Add-Printer -Name "sharpmx3070n" -DriverName "SHARP MX-3070N PCL6" -PortName
"printer3"

- name: black and white printing
  win_shell: Set-PrintConfiguration -PrinterName "sharpmx3070n" -Color 0 # B&W

- name: single-sided printing
  win_shell: Set-PrintConfiguration -PrinterName "sharpmx3070n" -DuplexingMode 'OneSided'
```

Ansible-Playbook PL

printer_install.yml

```
---
- name: Add CUPS printer for current site only
  hosts: windows_targets

  tasks:
    - name: Detect site by local IPv4
      ansible.windows.win_powershell:
        script: |
          $ips = Get-NetIPAddress -AddressFamily IPv4 | Where-Object {
            $_.IPAddress -notmatch '^169\.254\.' -and $_.IPAddress -notmatch '^127\.'
          } | Select-Object -ExpandProperty IPAddress
```

```
          $site = 'NONE'
          foreach ($ip in $ips) {
            if ($ip -match '^192\.168\.200\.\d+$') { $site = 'HWS'; break }
            if ($ip -match '^10\.53\.1\.\d+$')     { $site = 'FBS'; break }
            if ($ip -match '^10\.26\.1\.\d+$')     { $site = 'SPS'; break }
          }

          @{site=$site; ips=$ips} | ConvertTo-Json -Compress
        register: netinfo
        changed_when: false

      - name: Save site_info (parsed)
        ansible.builtin.set_fact:
          site_info: "{{ (netinfo.output | default([])) | join('') | from_json }}"

      - name: Save site_id
        ansible.builtin.set_fact:
          site_id: "{{ site_info.site | default('NONE') }}"



      - name: Install printers for Standort HWS
        ansible.builtin.include_tasks: printer_HWS.yml
        when: site_id == 'HWS'

      - name: Install printers for Standort FBS
        ansible.builtin.include_tasks: printer_FBS.yml
        when: site_id == 'FBS'

      - name: Install printers for Standort SPS
        ansible.builtin.include_tasks: printer_SPS.yml
        when: site_id == 'SPS'
```

löst zb printer_HWS.yml aus:

printer_HWS.yml

```
- name: Install printers for HWS
  block:
    - name: Ensure Internet Printing Client is enabled (best-effort)
```

```
      ansible.windows.win_powershell:
        script: |
          $candidates = @('Printing-InternetPrinting-Client','Printing-Internet Printing-
Client')
          foreach ($c in $candidates) {
            $f = Get-WindowsOptionalFeature -Online -FeatureName $c -ErrorAction
SilentlyContinue
            if ($f) {
              if ($f.State -ne 'Enabled') { Enable-WindowsOptionalFeature -Online -FeatureName
$c -All -NoRestart | Out-Null }
              break
            }
          }
          '{"ok":true}'
      changed_when: false
      failed_when: false

    - name: Add all printers (force) with Microsoft IPP Class Driver
      ansible.windows.win_powershell:
        script: |
          $PrinterName = "{{ item.printer_name }}"
          $PrinterURL  = "{{ item.printer_url }}"
          $DriverName  = "{{ item.ipp_driver }}"
          try {
            Add-Printer -Name $PrinterName -PortName $PrinterURL -DriverName $DriverName -
ErrorAction Stop
            '{"changed":true,"action":"Add-Printer","name":' + (ConvertTo-Json $PrinterName -
Compress) + '}'
          } catch {
            '{"changed":false,"action":"ignored","name":' + (ConvertTo-Json $PrinterName -
Compress) +
            ',"error":' + (ConvertTo-Json $_.Exception.Message -Compress) + '}'
          }
      loop: "{{ printers }}"
      register: add_results
      changed_when: (item.output[0] | default('{}') | from_json).changed | default(false)
      failed_when: false

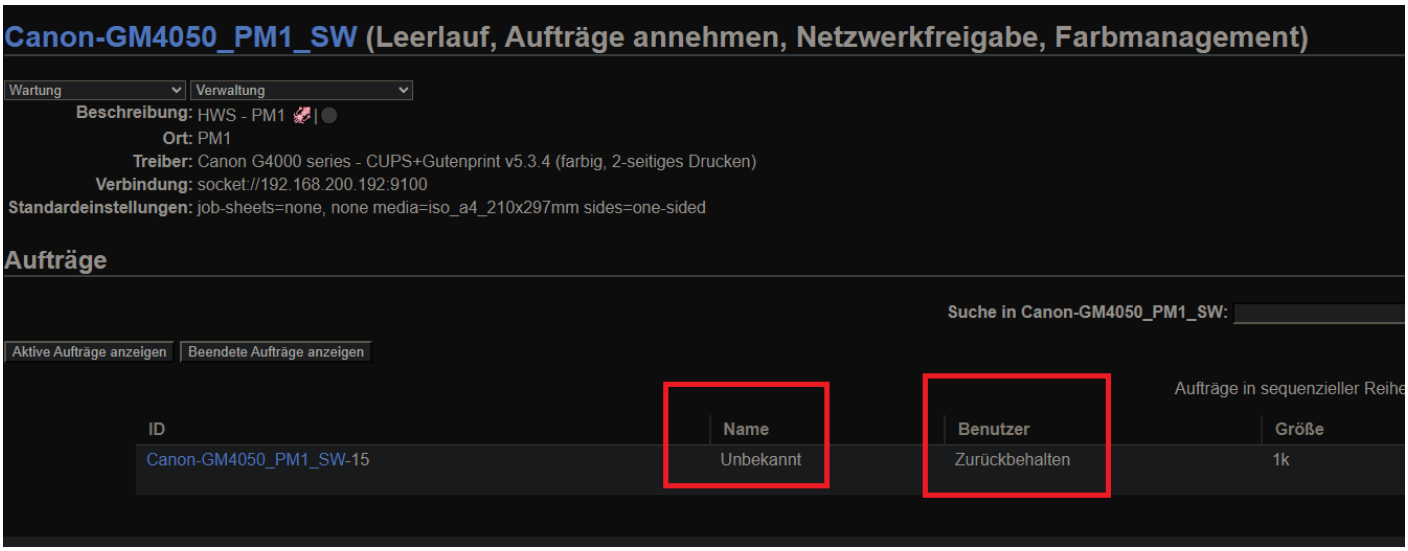
    - name: Summary (per printer)
      ansible.builtin.debug:
```

```
        msg: "{{ ((item.output | first) | default('{}')) | from_json }}"
      loop: "{{ add_results.results | default([]) }}"

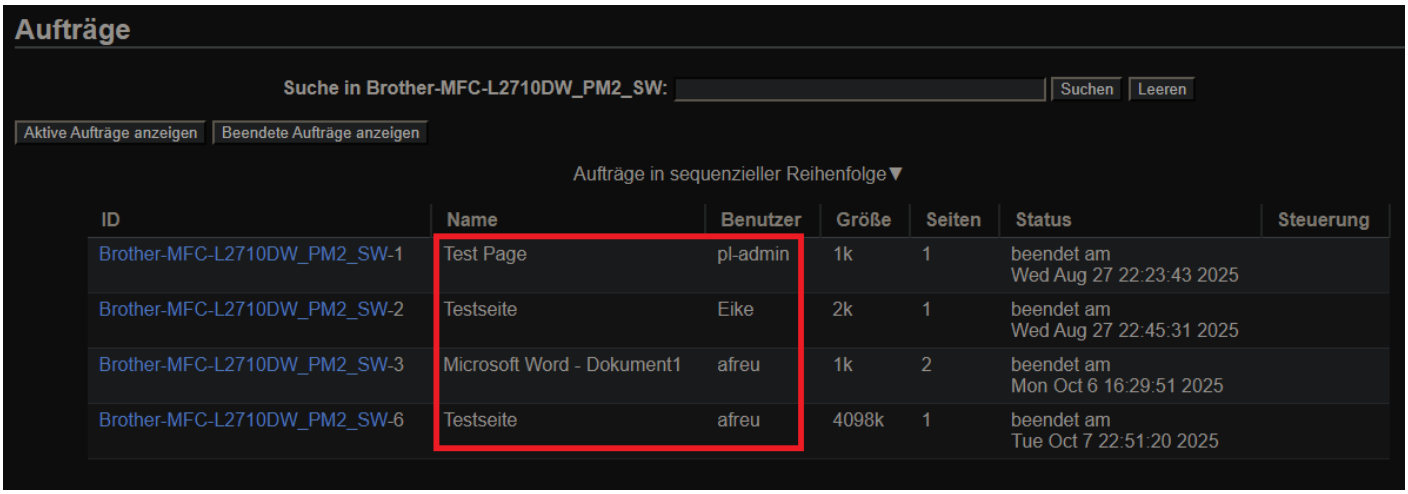

  vars:
    printers:
      - { printer_name: "HWS PM1 ∗□|□", printer_url:
"http://192.168.200.11:631/printers/Brother-MFC-L2750DW_PM1_SW", ipp_driver: "Microsoft IPP
Class Driver" }
      - { printer_name: "HWS PM1 □□□", printer_url: "http://192.168.200.11:631/printers/Canon-
GM4050_PM1_SW", ipp_driver: "Microsoft IPP Class Driver" }
      - { printer_name: "HWS PM2 ∗□|□", printer_url:
"http://192.168.200.11:631/printers/Brother-MFC-L2710DW_PM2_SW", ipp_driver: "Microsoft IPP
Class Driver" }
      - { printer_name: "HWS PM3 □□□", printer_url: "http://192.168.200.11:631/printers/Canon-
GM4050_PM3_SW", ipp_driver: "Microsoft IPP Class Driver" }
      - { printer_name: "HWS 1.OG ∗□|□□", printer_url:
"http://192.168.200.11:631/printers/Brother-DCP-L3560CDW_1.OG_Farbe", ipp_driver: "Microsoft
IPP Class Driver" }
```

Name und Benutzer anzeigen

Wenn man Name des Auftrags und Benutzer in der CUPS Web-UI sehen möchte und anstatt dem hier...

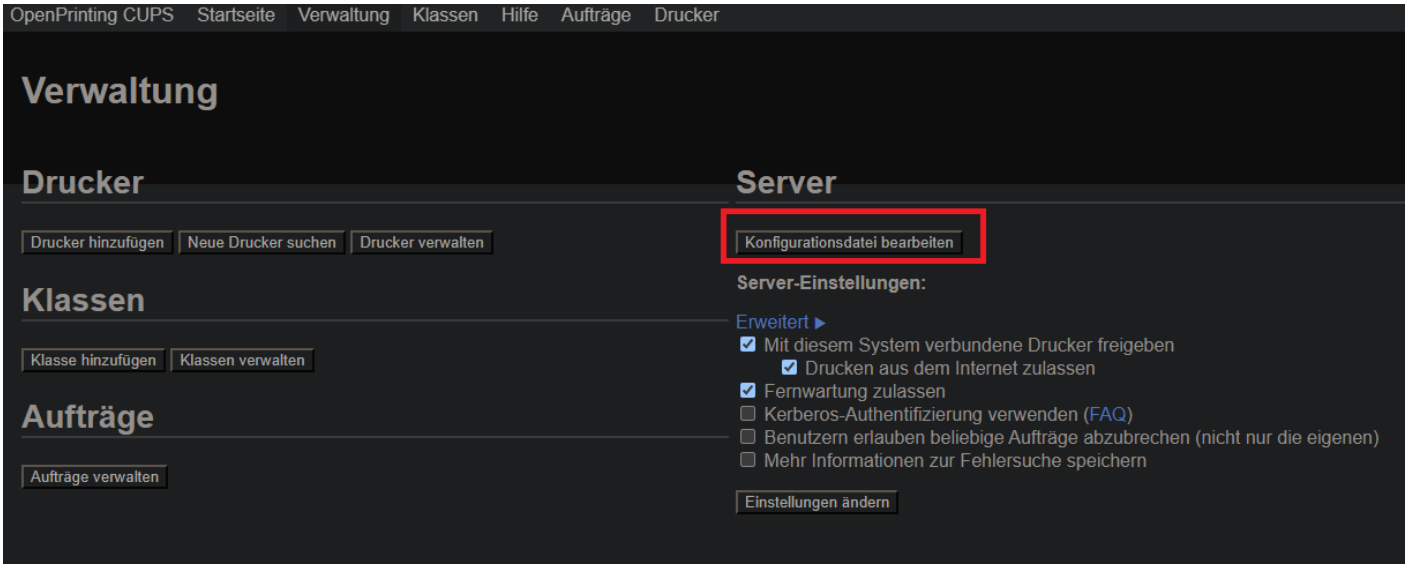


... lieber das hier sehen möchte ....



... muss die Konfigurationsdatei cupsd.conf wie folgt ändern:

Entweder direkt im Dockervolume oder CUPS Web-UI > Konfigurationsdatei bearbeiten:



folgende Stelle finden:

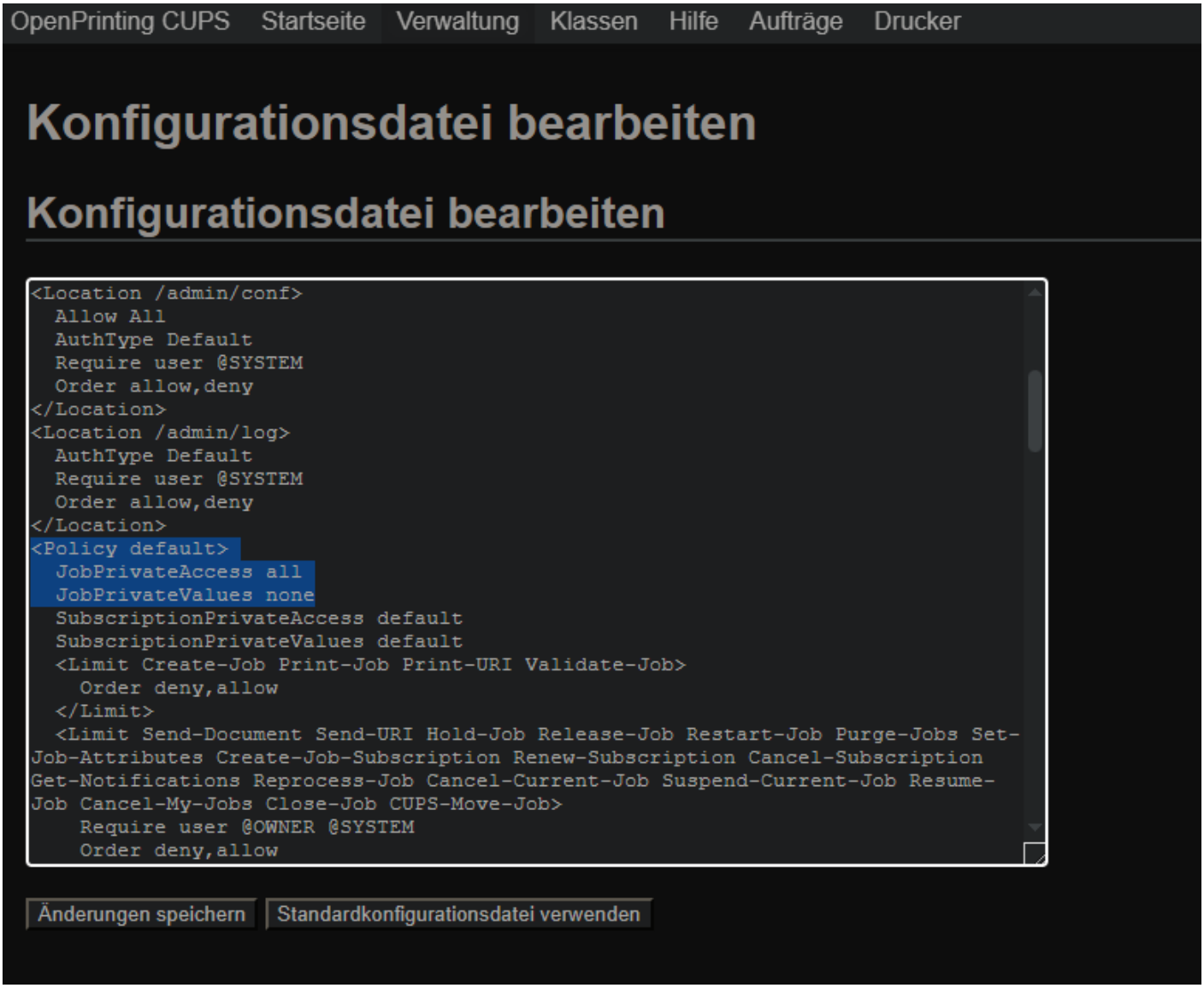
```
...
<Policy default>
  JobPrivateAccess default
  JobPrivateValues default
...
```

und in

```
...
<Policy default>
  JobPrivateAccess all
  JobPrivateValues none
```

```
...
```

ändern:



Änderungen speichern klicken und neustart abwarten. Dann unter Drucker > Auftäge nachsehen ob es geklappt hat.

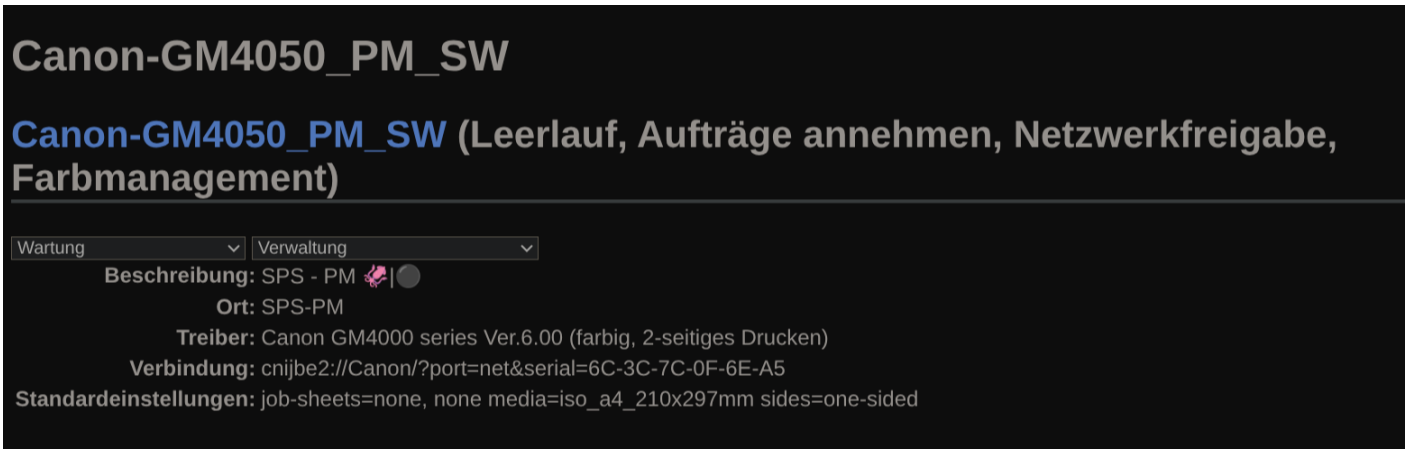


Canon Pixma GM4050: Keine Netzwerkfreigabe

Problem: Ein Canon Drucker meldet in CUPS, dass keine Netzwerkfreigabe möglich ist, obwohl das Häkchen gesetzt wurde:

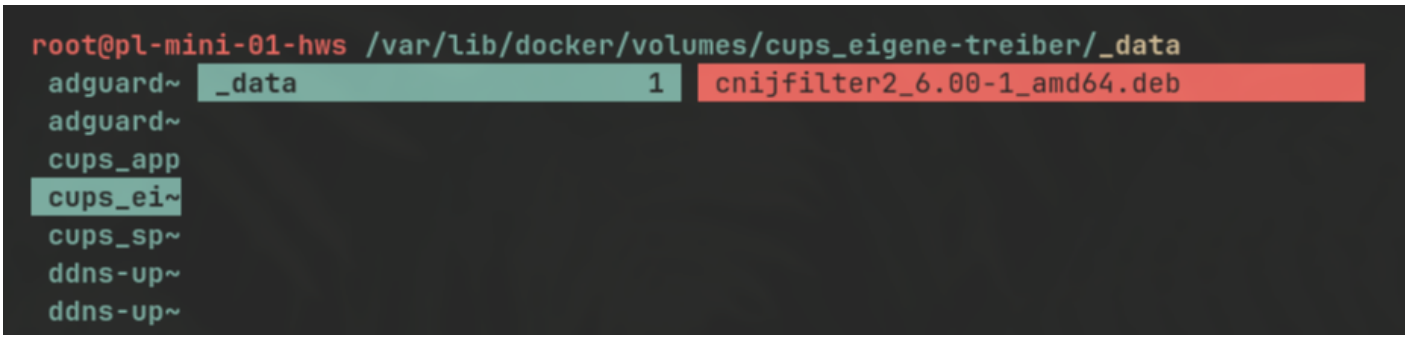


Lösung: in docker-compose den network_mode: host einstellen und den Drucker ueber CUPS > Verwaltung > Neuen Drucker suchen. Am besten vorher die Treiber installieren (siehe unten) damit entweder ueber socket:// oder cnijbe2:// (Canontreiber) verbunden werden kann:



Canon PIXMA GM4050: Druckertreiber nachinstallieren

1. Druckertreiber von offizieller Seite herunterladen, fuer Linux x64 (.deb oder .rpm)
2. eigenes Volume dafuer einbinden (siehe oben Docker Compose)
3. .deb in volume kopieren
4. command hinzufuegen um nach jeden Container-Start die Treiber zu installieren



docker-compose.yaml

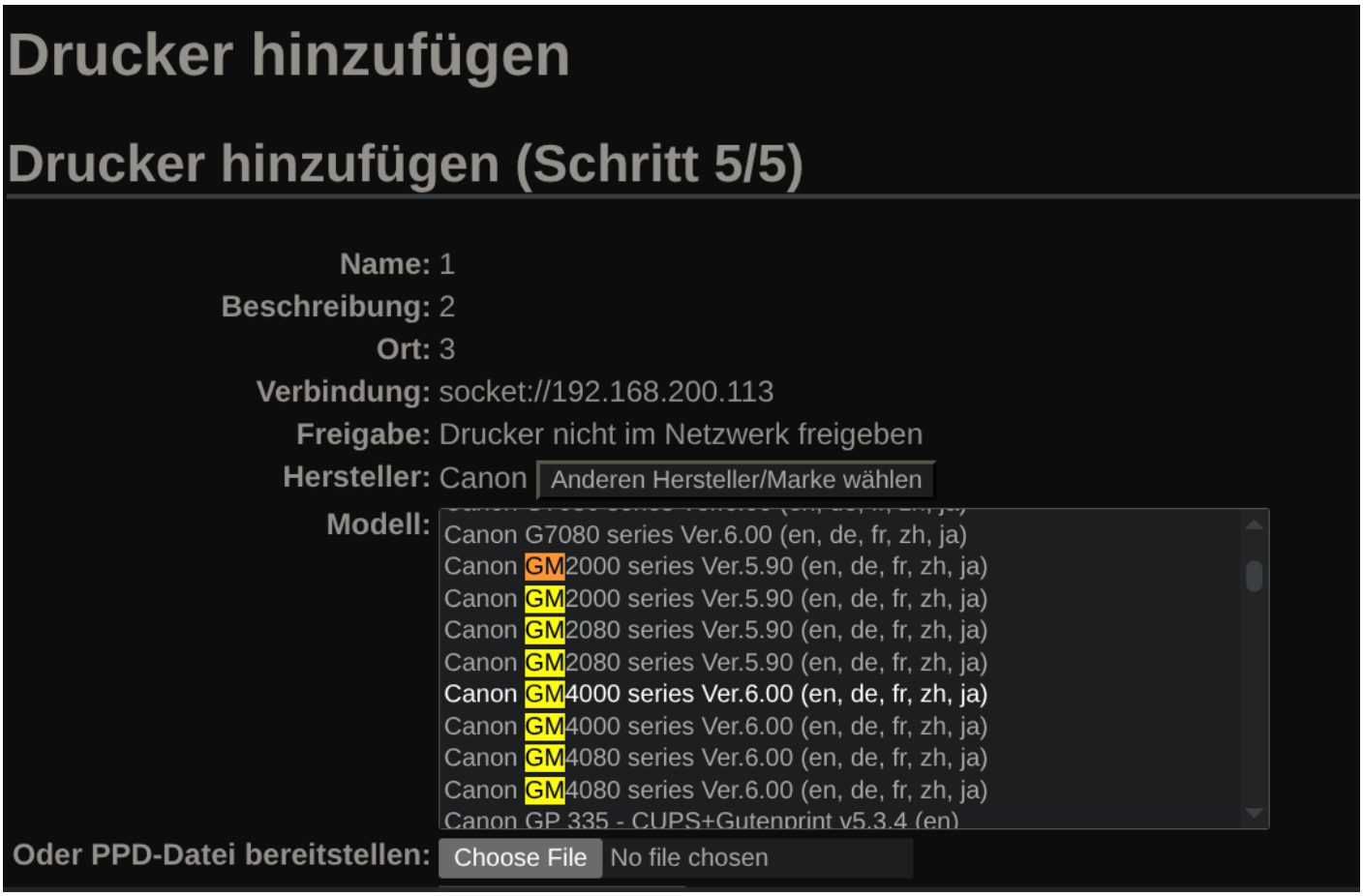
```
[...]
            - TZ=Europe/Berlin
        volumes:
            - app:/etc/cups
            - eigene-treiber:/opt/eigene-treiber
        command: sh -c "dpkg -i /opt/eigene-treiber/cnijfilter2_6.00-1_amd64.deb && apt-get
install -f -y && /entrypoint.sh"

volumes:
  app:
  eigene-treiber:
```

Drucker mit Socket hinzufuegen:

```
socket://<DRUCKER-IP>
```

Treiber auswaehlen:



Druckertreiber installieren

Powershell-CMDs
Treibername aus .inf Dateien herausfinden, nach Pattern filtern (GM4000 in diesem Fall für den GM4050 Drucker)
Get-Content "D:\Entwicklung\Praxis Maxis\Druckertreiber\Canon_GM4050*.inf" | Select-String -Pattern "GM4000" -Context 1,2

Druckertreiber installieren mit pnputil
pnputil /add-driver "D:\Entwicklung\Praxis Maxis\Druckertreiber\Canon_GM4050*.inf" /subdirs /install

Druckertreiber hinzufügen
Add-PrinterDriver -Name "Canon GM4000 series"

Drucker hinzufügen mit CUPS-Server in Docker und richtigem Treibername
Add-Printer -Name 'HWS PM3 □|●' -PortName 'http://192.168.200.11:631/printers/Canon-GM4050_PM3_SW' -DriverName 'Canon GM4000 series'

Falls ein Fehler auftaucht, probieren den Spooler-Service bei jedem Windows-Neustart neuzustarten mit
Restart-Service -Name Spooler -Force

Revision #21
Created 2025-08-27 21:57:32 UTC
Updated 2025-10-27 20:33:14 UTC