

IP Symcon als Container auf einem Raspi

Montag, 30. Mai 2022 15:42

1. System updaten (immer erster Schritt vor quasi allem)
 - o sudo apt update
 - o sudo apt upgrade
 - o sudo rpi-update
 - o sudo reboot
2. Docker installieren
 - o curl -sSL <https://get.docker.com> | sh
 - o sudo usermod -aG docker \$USER
 - o newgrp docker
3. Portainer installieren
 - o sudo docker pull portainer/portainer-ce:linux-arm
 - o sudo docker run --restart always --name=portainer -d -p 9443:9443 -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:linux-arm --http-disabled

Nach kurzer Zeit ist Portainer bereit und ihr könnt das Webinterface über den HTTPS-Port 9443 und die IP des Rasperry Pis erreichen. Zum Beispiel: <https://192.168.180.100:9443>
Dort dann einen User anlegen (mehr siehe <https://www.antary.de/2021/09/20/raspberry-pi-docker-und-portainer/>)

4. Geeigneten Container von Docker-Hub auswählen
Auf <https://hub.docker.com/r/symcon/symcon/tags> einen Container auswählen der zu den Vorstellungen passt, i.d.R. ist es sowas wie symcon/symcon:6.2-199-armhf . Dazu sucht ihr oben auf der Seite nach "arm" oder "arm64" je nachdem ob ihr 32 oder 64-Bit-Betriebssystem auf dem Pi habt. Dann wählt ihr die passende Version aus.
5. Container erstellen

Ihr könntet jetzt mit einem Befehl wie:

```
docker run --rm \  
  --name symcon \  
  --hostname symcon \  
  -p 3777:3777 \  
  -v /opt/symcon/data:/var/lib/symcon \  
  -v /opt/symcon/log:/var/log/symcon \  
  -v /opt/symcon:/root \  
  symcon/symcon:6.2-199-armhf
```

zwar den Container schon erstellen und IP Symcon würde laufen und ihr würdet es in Portainer sehen können, aber euch würden wichtige Details fehlen, z.B. USB-Geräte. Außerdem ist es hilfreich dem Container beim runterfahren etwas mehr Zeit zu genehmigen. Aus diesem Grund mache ich es etwas komplizierter, aber versprochen, es lohnt sich....

Stacks in Portainer: Stacks ist die Portainer-Variante von Docker-Compose (und Docker-Compose ist eine Beschreibungssprache wie ein Container aussehen soll).

In der linken Navigation Stacks auswählen +Add stack anklicken und bei Name etwas eingeben wie ipsymcon. Danach auf Web-Editor klicken und im Eingabefeld folgendes rein pasten (Achtung das nichts verrutscht, Einrückungen sind wichtig!):

```
version: "3"  
services:  
  Symcon:  
    cap_add:  
      - AUDIT_WRITE  
      - CHOWN  
      - DAC_OVERRIDE  
      - FOWNER  
      - FSETID  
      - KILL  
      - MKNOD  
      - NET_BIND_SERVICE  
      - NET_RAW  
      - SETFCAP  
      - SETGID  
      - SETPCAP  
      - SETUID  
      - SYS_CHROOT  
    cap_drop:  
      - AUDIT_CONTROL  
      - BLOCK_SUSPEND  
      - DAC_READ_SEARCH  
      - IPC_LOCK  
      - IPC_OWNER  
      - LEASE  
      - LINUX_IMMUTABLE  
      - MAC_ADMIN  
      - MAC_OVERRIDE  
      - NET_ADMIN  
      - NET_BROADCAST  
      - SYSLOG  
      - SYS_ADMIN
```

```
- NET_ADMIN
- NET_BROADCAST
- SYSLOG
- SYS_ADMIN
- SYS_BOOT
- SYS_MODULE
- SYS_NICE
- SYS_PACCT
- SYS_PTRACE
- SYS_RAWIO
- SYS_RESOURCE
- SYS_TIME
- SYS_TTY_CONFIG
- WAKE_ALARM
```

command:

```
- /usr/bin/symcon
```

container_name: Symcon

devices:

```
- /dev/ttyUSB0:/dev/ttyUSB0
```

```
- /dev/ttyUSB1:/dev/ttyUSB1
```

environment:

```
-
```

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

```
- DOCKER=true
```

```
- DEBIAN_FRONTEND=noninteractive
```

```
- TZ=Europe/Berlin
```

```
- LANGUAGE=de_DE.utf8
```

```
- LC_ALL=de_DE.utf8
```

```
- LANG=de_DE.utf8
```

image: symcon/symcon:6.3-202-armhf

ipc: private

logging:

```
driver: json-file
```

```
options: {}
```

network_mode: "host"

restart: always

stop_grace_period: 60s

volumes:

```
- /opt/symcon:/root
```

```
- /opt/symcon/data:/var/lib/symcon
```

```
- /opt/symcon/log:/var/log/symcon
```

Hier sind die USB-Geräte für den Container freigegeben worden. Wenn ihr mehr oder andere habt, nach diesem Muster verändern.

Ich verwende das Netzwerk des Host direkt, was Freigaben von Ports überflüssig macht. Ist nicht die ideale Lösung, aber insb. für Anfänger die mit den geringsten Fehlern und entspricht der nativ installierten IPS-Version.

Hier wird definiert wo die Daten auf dem realen Pi liegen, die im Container woanders liegen. Also /opt/symcon/data ist z.B. das Verzeichnis auf dem Pi und das nach dem :, also /var/lib/symcon ist das Verzeichnis IM Container, also wo z.B. bei einer nativen Installation auch die Daten liegen würden.

Wenn du also am Pi direkt angemeldet bist, wirst du unter /opt/symcon/log die Log-Dateien einsehen können.

Wenn ihr jetzt auf "deploy the stack" drückt, sollte ganz kurz Zeit später unter der IP:3777 euer IP-Symcon aufzufinden sein... denn dadurch wird das Script, was ihr später immer noch anpassen könnt, gespeichert, der Container erstellt, ggf. das Container-Image geholt und gestartet in einem Zuge.

Alle (Backup-)relevanten Daten befinden sich unterhalb von /opt/symcon und können so einfach gesichert werden.