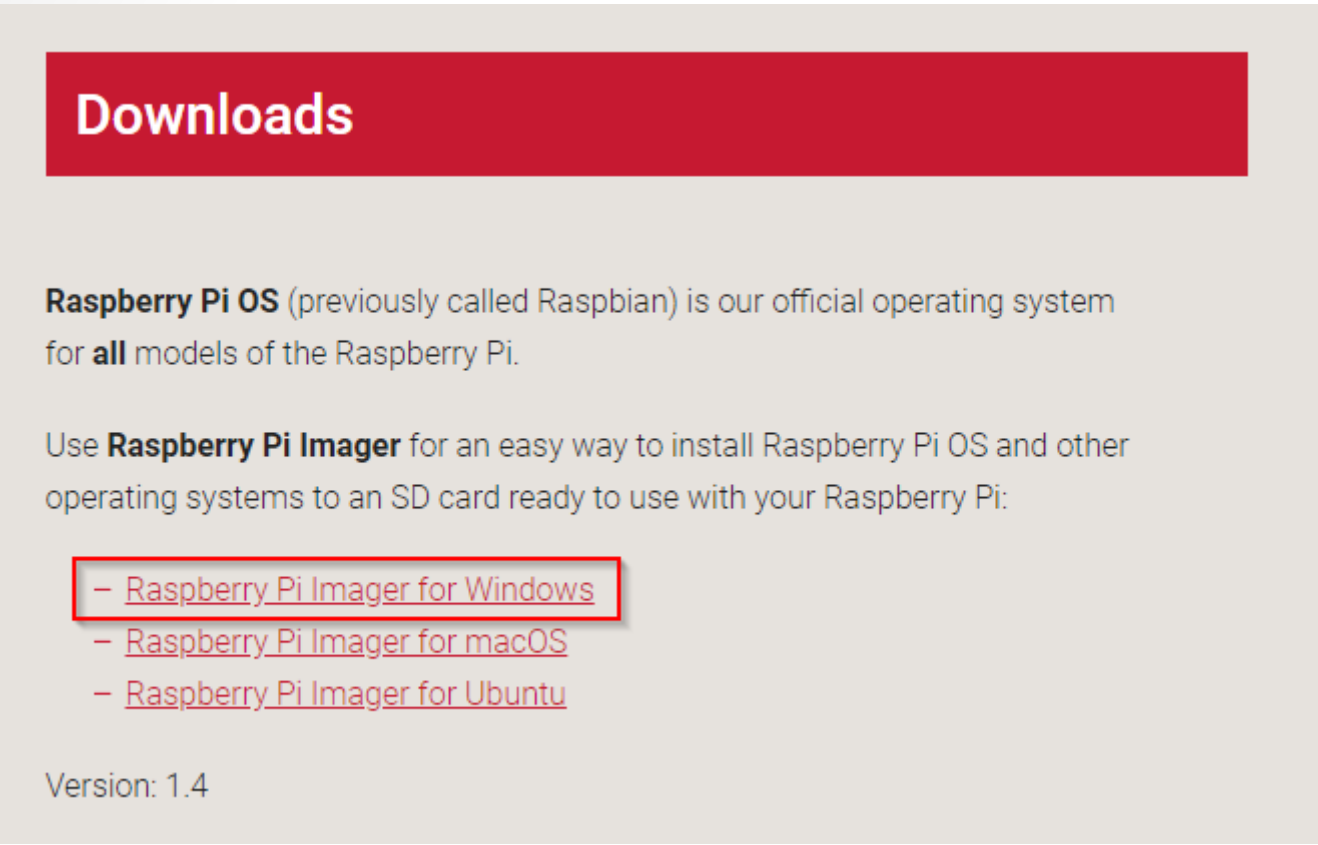
Raspberry Pi auf Sd-Card installieren

- Installation über Imager
- https://www.raspberrypi.org/downloads/

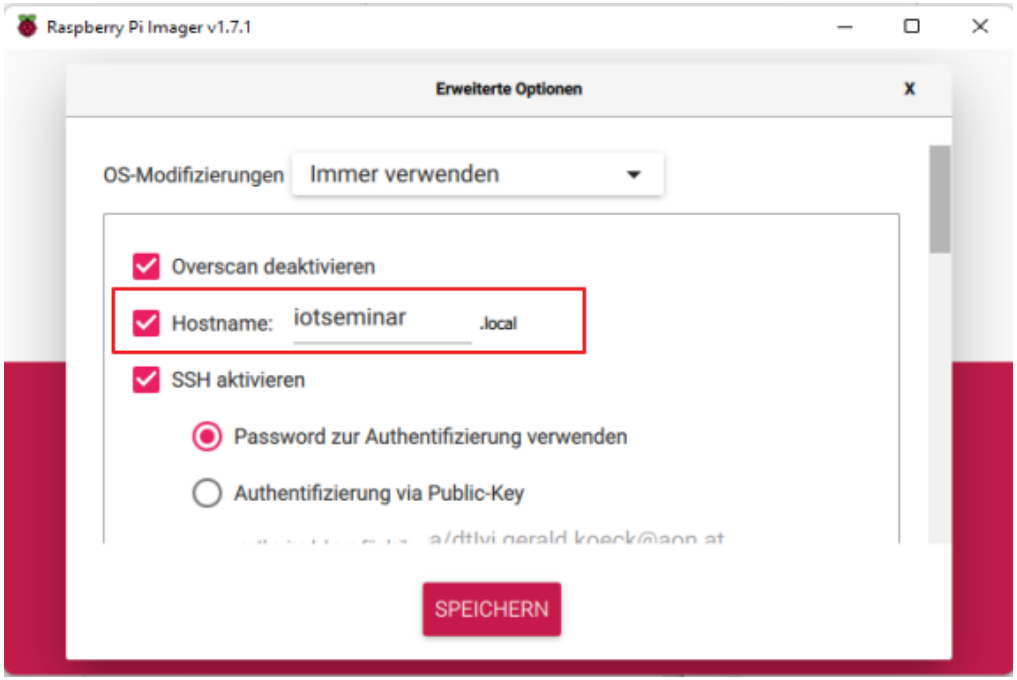


Imager anwenden

- OS und SD-Card wählen und starten
- Raspberry Pi OS Lite genügt
  - Seit 2022 64Bit-Version

Neues Tool ➔ Vorkonfiguration - scrollen

- Strg-Shift-x ➔ zusätzliches Menü
  - Hostname auf eigenen Namen setzen
  - pi-Passwort für Seminar iot20220314
  - Wifi-Credentials iot/iot20220314
  - Zeitzone
  - Tastaturlayout
  - …



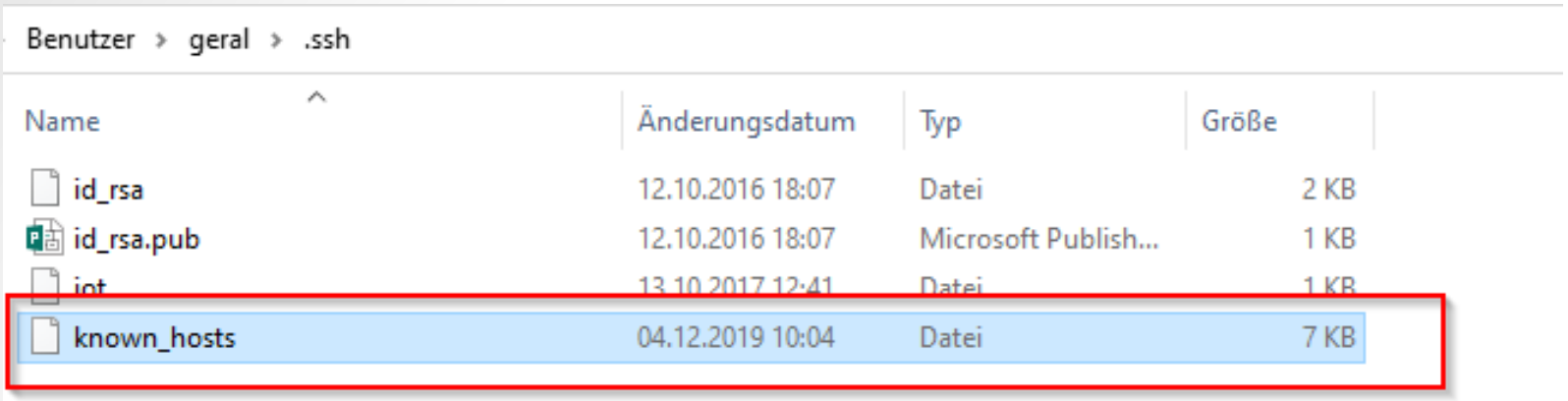
Erstinbetriebnahme

- Raspi starten und einige Minuten warten
- Versuch Raspi über SSH zu erreichen
  - ssh pi@iotseminar.local
  - ifconfig ➔ IP V4 Adresse
- IP-Adresse ausfindig machen
  - Router ➔ Netzwerk ➔ Raspi suchen
- Mit ssh pi@192... verbinden
- Raspi aktualisieren
  - sudo apt update && sudo apt full-upgrade
- sudo reboot now

Raspi war unter der IP schon im Netz

```
PS D:\work\IOT\IotSamstag> ssh pi@192.168.0.105
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:EcdcRitiAenorXjWMqZGgjfoyzOYvtwsxRD2Xk3oCNw.
Please contact your system administrator.
Add correct host key in C:\\Users\\geral/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in C:\\Users\\geral/.ssh/known_hosts:35
ECDSA host key for 192.168.0.105 has changed and you have requested strict checking.
Host key verification failed.
```

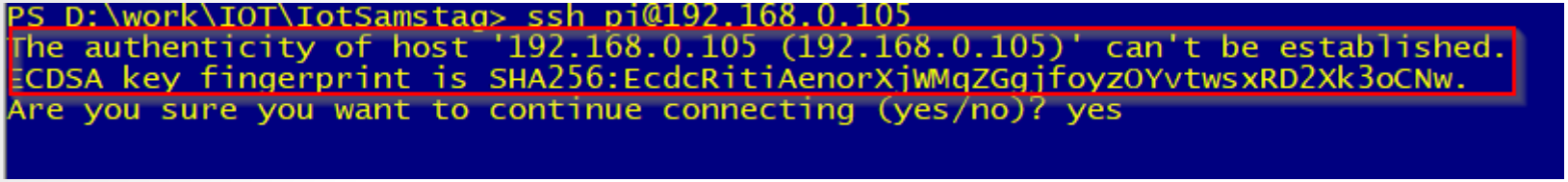
Aus known_hosts entfernen



known_hosts

```
33  192.168.0.89 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGRj1QeSHMNSj2lYNn/x6MkyKFjuwrb7ud1niGvaPvxw0b
34  192.168.0.108 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDu2mtOlhq5KbbJQYVQgH3WLNhofFdmUB9OHOIRFSreA1
35  192.168.0.105 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKFPtIEuNkj5HUrIg9gS1cLJ1UDVkkmDZA28YAVT9kzJA
36  192.168.0.56 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKFPtIEuNkj5HUrIg9gS1cLJ1UDVkkmDZA28YAVT9kzJAM
37  192.168.0.58 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBBjPE42ur2+x2cudXUxsDsASuUpxG6zHlDoLaU0sv3Fbzb
38
```

Wird gleich wieder eingetragen



- Einstiegsaccount pi/raspberry

Bootloader bei Bedarf aktualisieren

```
pi@raspberrypi4:~ $
pi@raspberrypi4:~ $ sudo rpi-eeprom-update
*** UPDATE AVAILABLE ***
BOOTLOADER: update available
   CURRENT: Do 3. Sep 12:11:43 UTC 2020 (1599135103)
    LATEST: Do 29. Apr 16:11:25 UTC 2021 (1619712685)
   RELEASE: default (/lib/firmware/raspberrypi/bootloader/default)
            Use raspi-config to change the release.

  VL805_FW: Dedicated VL805 EEPROM
     VL805: up to date
   CURRENT: 000138a1
    LATEST: 000138a1
pi@raspberrypi4:~ $ sudo rpi-eeprom-update -a
*** INSTALLING EEPROM UPDATES ***

BOOTLOADER: update available
   CURRENT: Do 3. Sep 12:11:43 UTC 2020 (1599135103)
    LATEST: Do 29. Apr 16:11:25 UTC 2021 (1619712685)
   RELEASE: default (/lib/firmware/raspberrypi/bootloader/default)
            Use raspi-config to change the release.

  VL805_FW: Dedicated VL805 EEPROM
     VL805: up to date
   CURRENT: 000138a1
    LATEST: 000138a1
   CURRENT: Do 3. Sep 12:11:43 UTC 2020 (1599135103)
    UPDATE: Do 29. Apr 16:11:25 UTC 2021 (1619712685)
    BOOTFS: /boot

EEPROM updates pending. Please reboot to apply the update.
To cancel a pending update run "sudo rpi-eeprom-update -r".
pi@raspberrypi4:~ $ sudo reboot
Connection to 172.16.0.160 closed by remote host.
Connection to 172.16.0.160 closed.
```

WERBEVIDEO

! Solltest Du Version 2020-09-03 (oder später) haben, ist nichts weiter zu tun. Ich zeige trotzdem, wie man die Version aktualisiert.

HINWEIS

WLAN konfigurieren

- wpa_supplicant.conf in boot-Folder anlegen
- Funktioniert nicht immer
  - Falsche Zeichen in SSID?

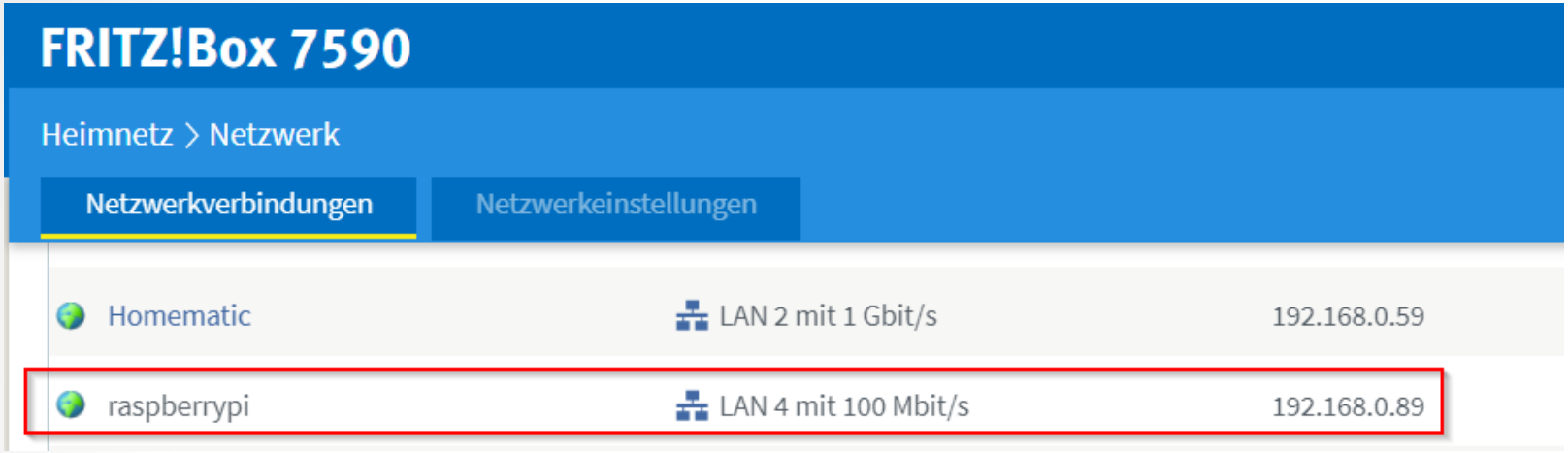
```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=AT

network={
        ssid="linksysAtHome"
        psk="xxx"
}

network={
        ssid="A1-B5035B"
        psk="xxx"
}
```

Wenn headless Booten über WLAN scheitert

- Kabel anstecken
- Router nach Raspi suchen

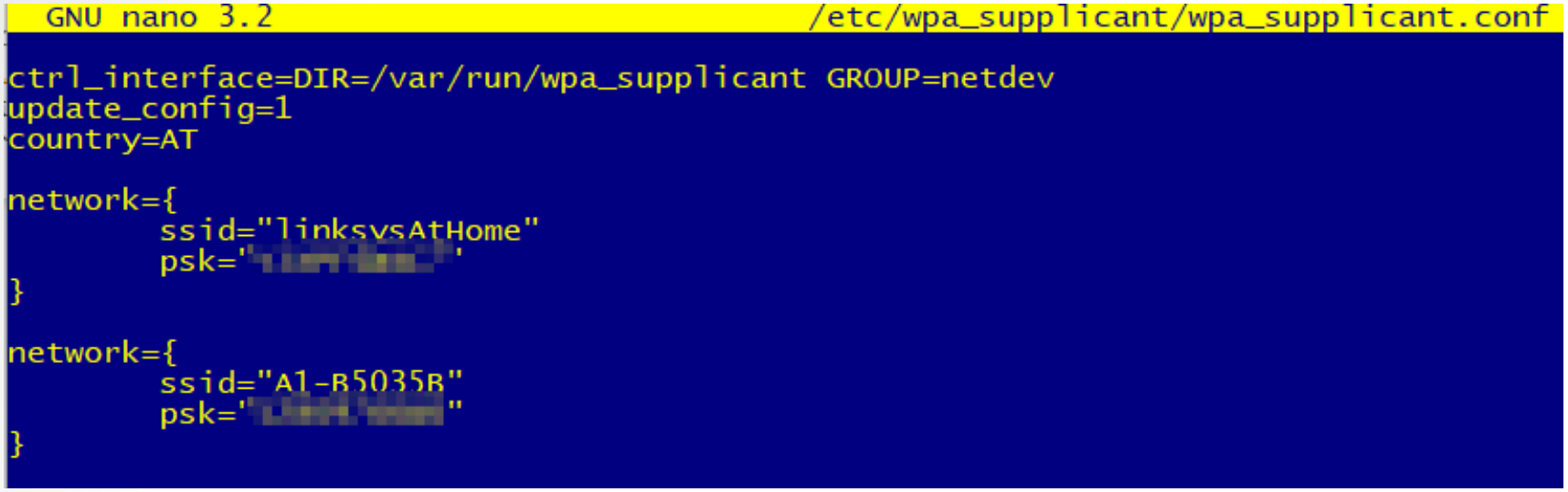


sudo raspi-config aufrufen

- Land für WLAN definieren
- WLAN einrichten
  - Bei Bedarf mehrere WLANs einrichten

WLAN-Konfiguration überprüfen

- sudo nano /etc/wpa_supplicant/wpa_supplicant.conf



Samba installieren

- https://www.elektronik-kompendium.de/sites/raspberry-pi/2007071.htm
- Test, ob Samba läuft
  - sudo service smbd status
  - sudo service nmbd status
- Samba installieren
  - sudo apt-get install samba samba-common smbclient
- User pi anlegen: sudo smbpasswd -a pi
- Konfiguration: sudo nano /etc/samba/smb.conf
  - Testen: testparm
- Alles lesbar freigeben:

```
[root]
    comment = root readonly
    path = /
    read only = yes
    browsable = yes
```

- Vor Verbindung kontrollieren, ob nicht bereits eine alte Verbindung zu der IP-Adresse besteht ➔ entfernen

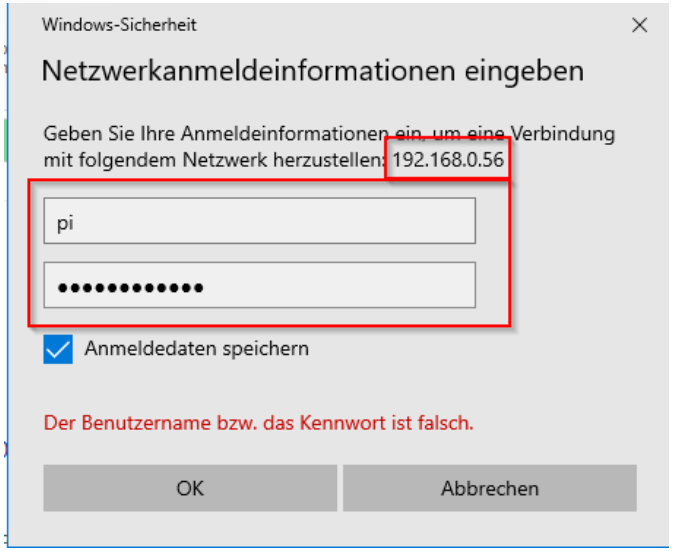
Beschreibbare Freigaben

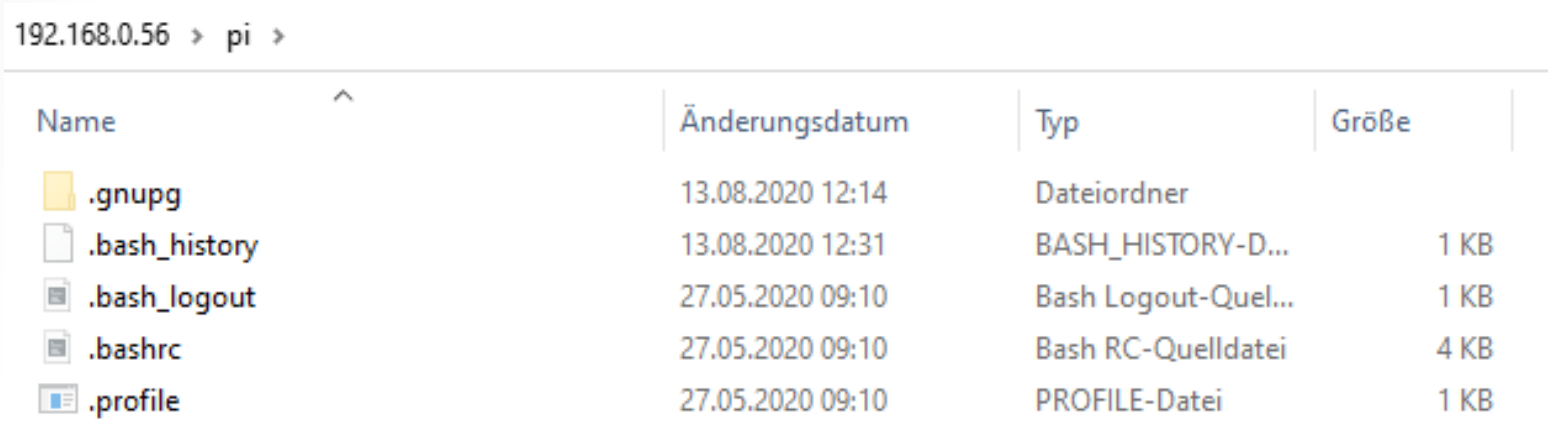
```
[opt]
    comment = opt
    path = /opt/
    writeable = yes
    browsable = yes

[pi]
    comment = home pi
    path = /home/pi/
    writeable = yes
    browsable = yes
```

- Restart
  - sudo service smbd restart

Samba testen





Git installieren

- sudo apt-get install git

Docker installieren

- Scriptverzeichnis in /home/pi/ anlegen
  - mkdir scripts
  - cd scripts
- Docker Installationsskript downloaden und ausführen
  - curl -fsSL https://get.docker.com -o get-docker.sh
  - sudo sh get-docker.sh ➔ dauert etwas
- Benutzer pi zur Dockergruppe hinzufügen
  - sudo usermod -aG docker pi
- Docker testen
  - sudo docker version
  - sudo docker info

docker-compose installieren

- Abhängigkeiten installieren
  - sudo apt-get install -y libffi-dev libssl-dev
  - sudo apt-get install -y python3 python3-pip
- docker-compose über Python installieren
  - sudo pip3 -v install docker-compose ➔ dauert

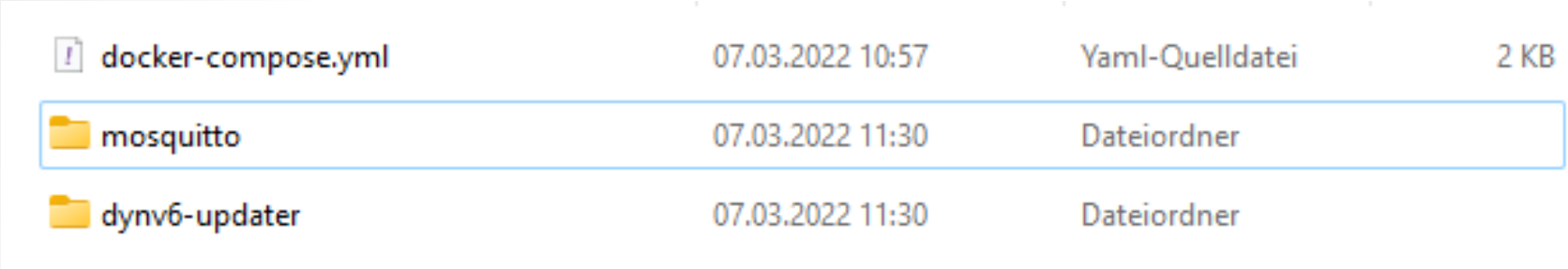
Service Portainer

pi > docker > docker-compose.yml

```yaml
version: "3.8"
services:
  portainer:
    image: portainer/portainer:latest
    container_name: portainer
    command: -H unix:///var/run/docker.sock
    restart: always
    ports:
      - 9000:9000
      - 8000:8000
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - portainer-data:/data
```

compose-Files strukturieren

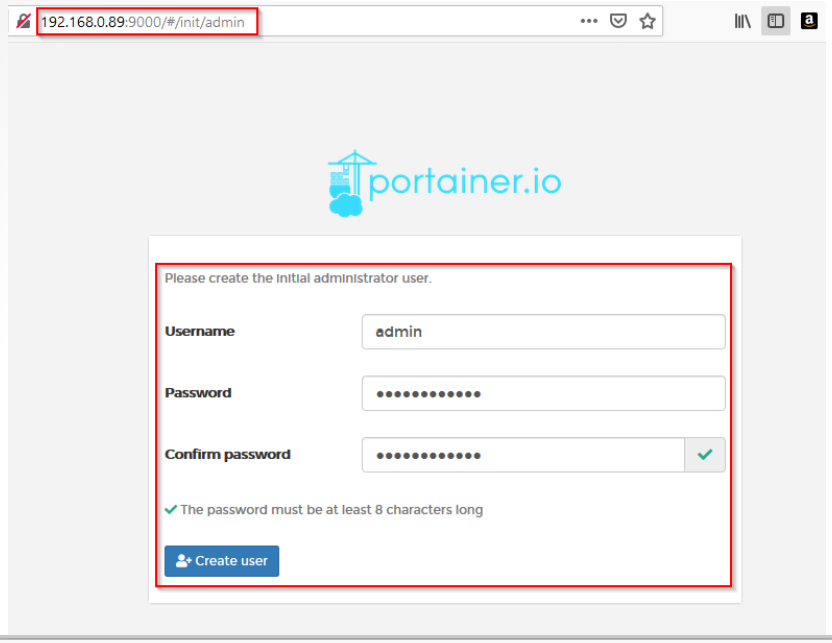
- Ein Compose-File mit allen Services
- Unterordner enthalten Volumes, wenn notwendig

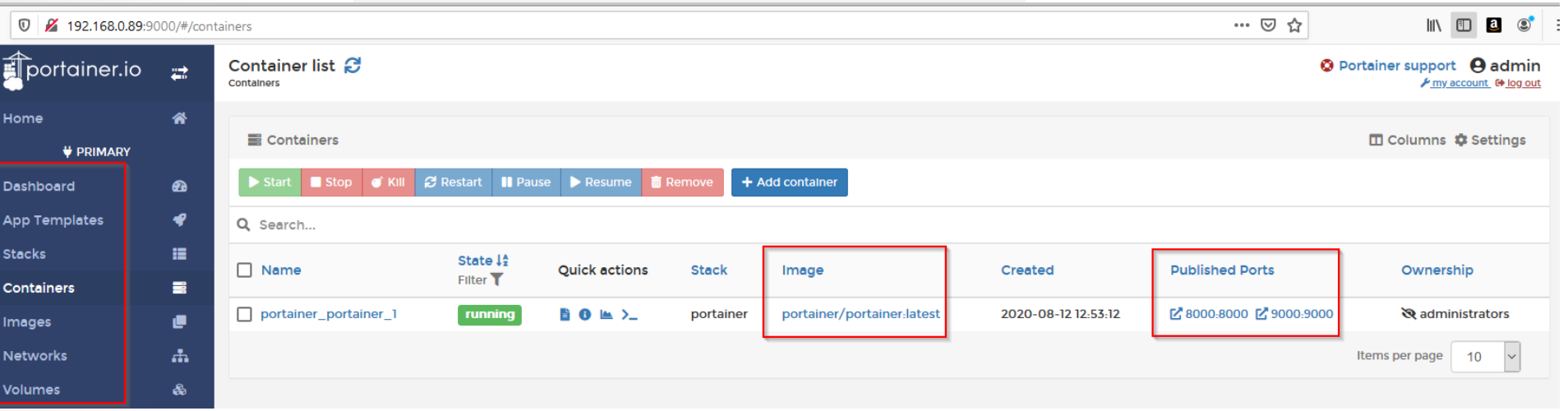


- Einzelnen Container starten
  - sudo docker-compose up [-d] portainer

Portainer läuft

- 192.168.0.56:9000

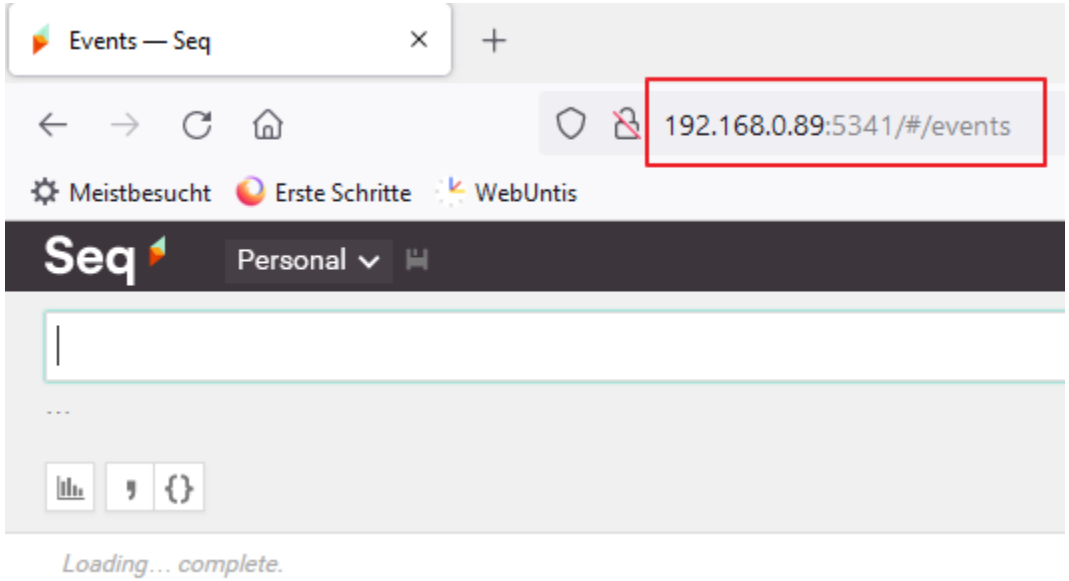




LogServer Seq

```yaml
version: '3'
services:
  seq:
    image: datalust/seq:2021.3.6681-arm64
    ports:
      - 5341:80
    environment:
      ACCEPT_EULA: Y
```

Deswegen armv8 (64bit!)



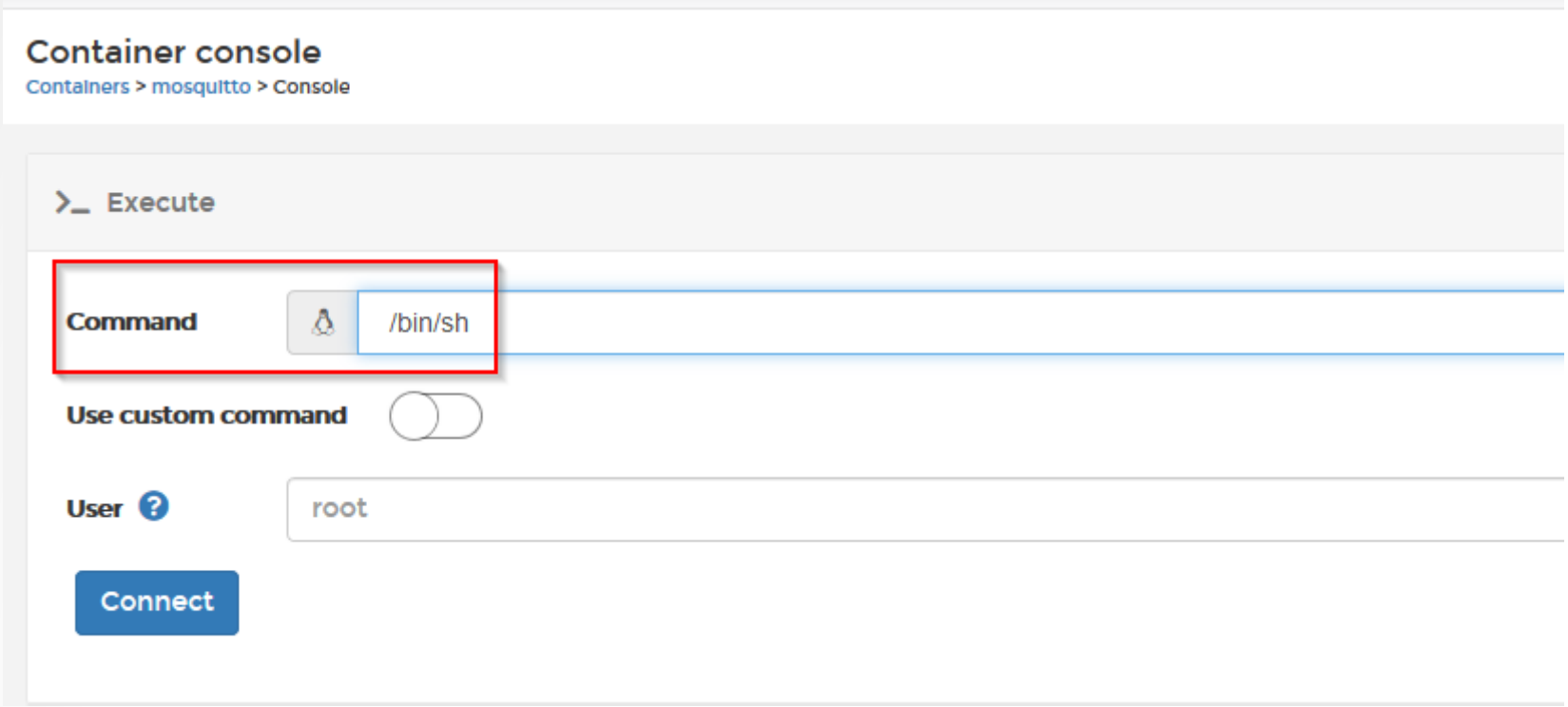
Mqtt-Broker Mosquitto

- Volumes auf Host gemappt ➔ Config

```
24  mqtt:
25    image: eclipse-mosquitto
26    container_name: mosquitto
27    restart: always
28    expose:
29      - 1883
30      - 8883
31      - 9001
32    ports:
33      - 1883:1883
34      - 8883:8883
35      - 9001:9001
36    volumes:
37      - ./mosquitto/config:/mosquitto/config
38      - ./mosquitto/log:/mosquitto/log
39      - ./mosquitto/data:/mosquitto/data
40      - /etc/localtime:/etc/localtime:ro
```

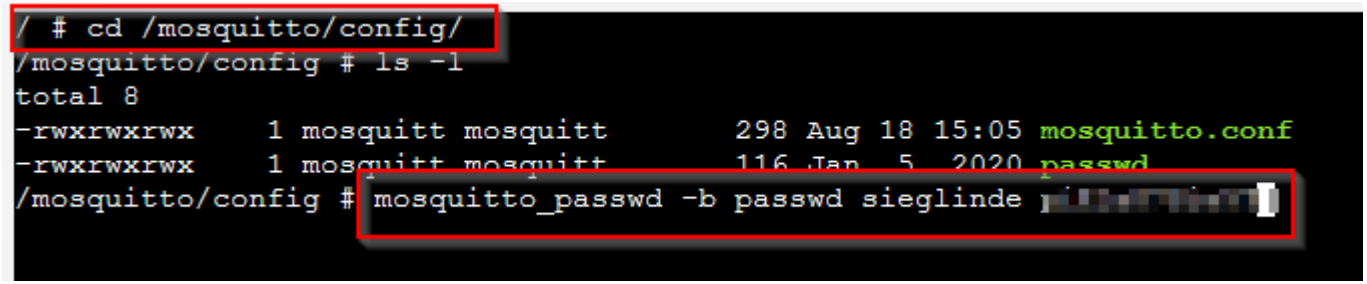
Passwort erzeugen

- Container ohne eigene Config starten
  - Oder in config mit `allow_anonymous true`
- Portainer ➔ Commandline shell



Commandline im Container

- Kennwort erzeugen
  - mosquitto_passwd -c passwordfile user
- Kennwort hinzufügen
  - mosquitto_passwd -b passwordfile user password
- RESTART CONTAINER



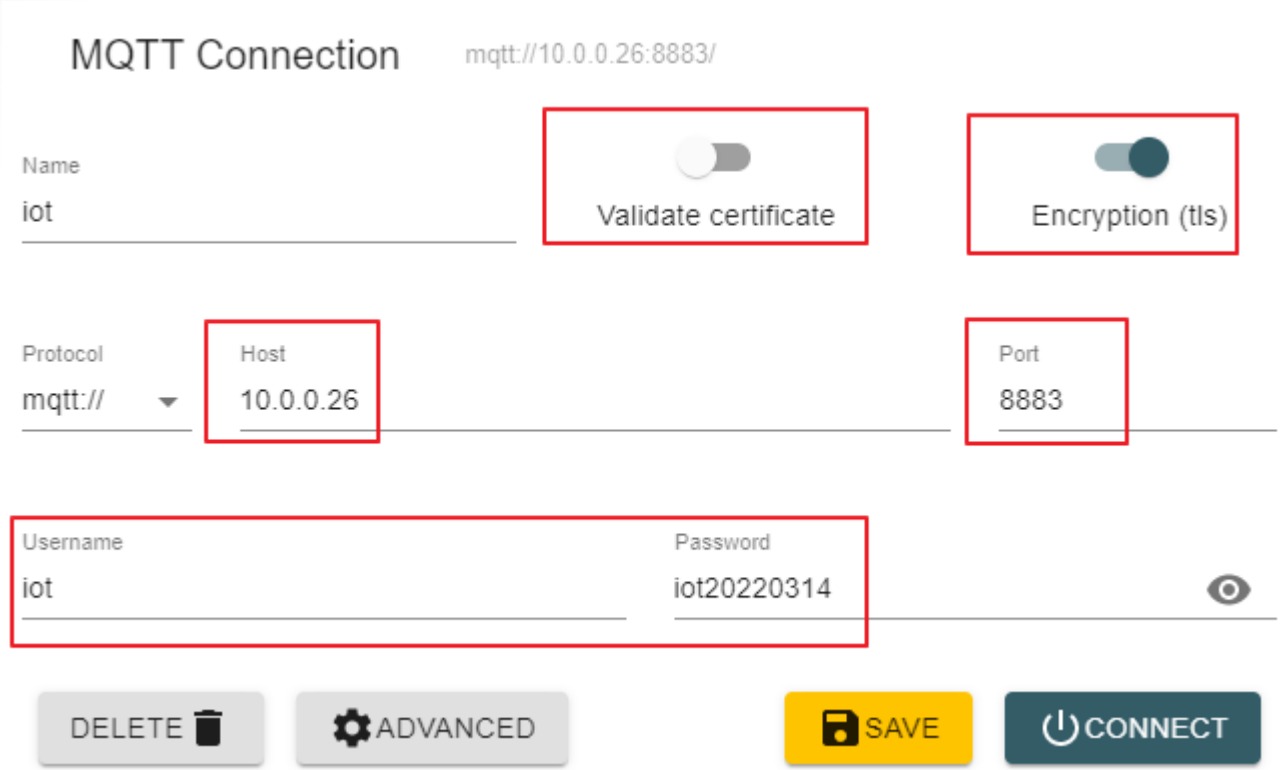
- Kennwort löschen
  - mosquitto_passwd -D passwordfile user

Kontrolle im File

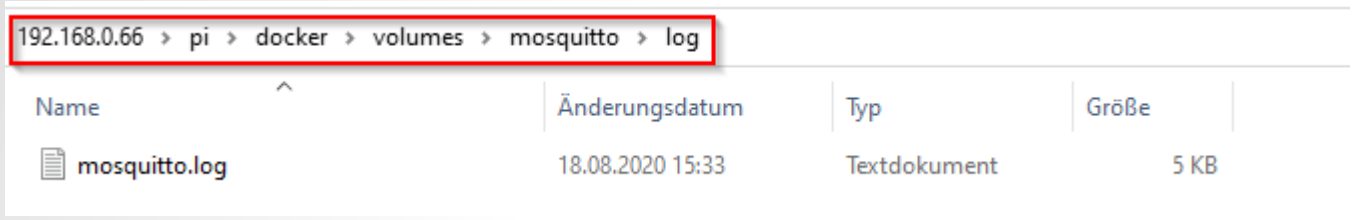
```
/mosquitto/config # cat passwd
gerald:$6$jk36WkI+kiIjx8zm$N7pBKLK/0CjCaH7+TEe71tQwtqWEWQmPIS6Sm9qoGZtxptIjDRqfxKCavtjz25ZtRPXvS0n0po3pnKepB6jfBA==
sieglinde:$6$K7bBDoGTUV+yqaBw$/EqFtL1EG37D59NI7VAePt1fxBpt7ELvJ90QiXXt6pvknPS1DIzWPDj7VoFucvQJK6MBd0dH1+U0eGZMrfJxKw==
/mosquitto/config #
```

Kontrolle des externen Zugriffs

- MQTT-Explorer
  - http://mqtt-explorer.com/



Zugriff auf Logfile über Volume



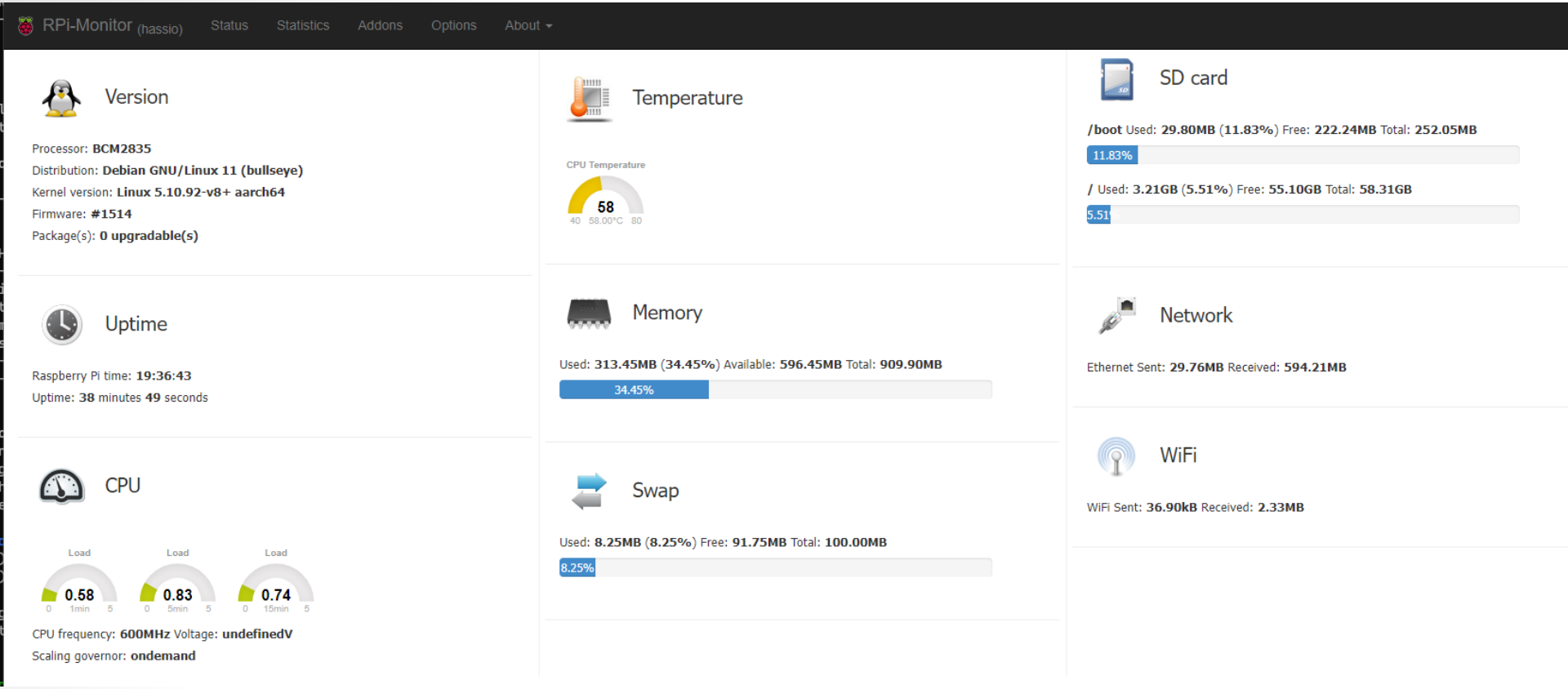
mosquitto.log

```
64   1597757598: mosquitto version 1.6.11 starting
65   1597757598: Config loaded from /mosquitto/config/mosquitto.conf.
66   1597757598: Opening ipv4 listen socket on port 1883.
67   1597757598: Opening ipv6 listen socket on port 1883.
68   1597757598: Opening ipv4 listen socket on port 8883.
69   1597757598: Opening ipv6 listen socket on port 8883.
70   1597757598: mosquitto version 1.6.11 running
71   1597757607: New connection from 194.166.174.203 on port 8883.
72   1597757607: New client connected from 194.166.174.203 as mqtt-explorer-8700ba4c (p2, c1, k60, u'sieglinde').
73
```

```
 GNU nano 3.2                                    mosquitto.log

1627983959: mosquitto version 2.0.11 starting
1627983959: Config loaded from /mosquitto/config/mosquitto.conf.
1627983959: Opening ipv4 listen socket on port 1883.
1627983959: Opening ipv6 listen socket on port 1883.
1627983959: Opening ipv4 listen socket on port 8883.
1627983959: Opening ipv6 listen socket on port 8883.
1627983959: mosquitto version 2.0.11 running
1627983959: New connection from 192.168.0.20:1956 on port 1883.
1627983959: New client connected from 192.168.0.20:1956 as mqtt-explorer-5d0a4a15 (p2, c1, k60, u'gerald').
```

RPI-Monitor


RPi-Monitor (hassio)
Status
Statistics
Addons
Options
About
Version
Processor: BCM2835
Distribution: Debian GNU/Linux 11 (bullseye)
Kernel version: Linux 5.10.92-v8+ aarch64
Firmware: #1514
Package(s): 0 upgradable(s)
Uptime
Raspberry Pi time: 19:36:43
Uptime: 38 minutes 49 seconds
CPU
Load
0.58
0 1min 5
Load
0.83
0 5min 5
Load
0.74
0 15min 5
CPU frequency: 600MHz Voltage: undefinedV
Scaling governor: ondemand
Temperature
CPU Temperature
58
40 58.00°C 80
Memory
Used: 313.45MB (34.45%) Available: 596.45MB Total: 909.90MB
34.45%
Swap
Used: 8.25MB (8.25%) Free: 91.75MB Total: 100.00MB
8.25%
SD card
/boot Used: 29.80MB (11.83%) Free: 222.24MB Total: 252.05MB
11.83%
/ Used: 3.21GB (5.51%) Free: 55.10GB Total: 58.31GB
5.51
Network
Ethernet Sent: 29.76MB Received: 594.21MB
WiFi
WiFi Sent: 36.90kB Received: 2.33MB

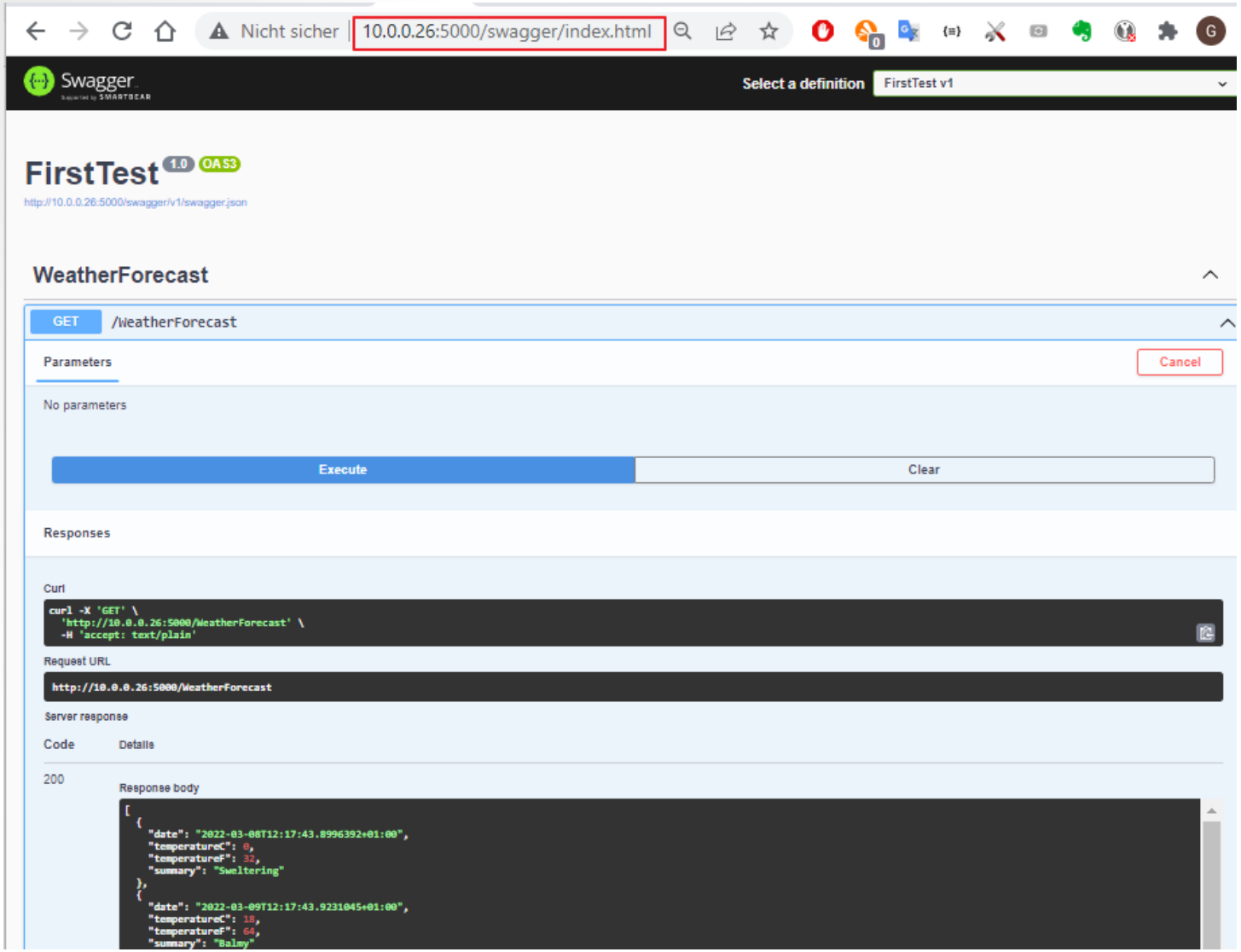
dotnet 6 installieren

- https://docs.microsoft.com/en-us/dotnet/iot/deployment
  - curl -sSL https://dot.net/v1/dotnet-install.sh | bash /dev/stdin --channel Current
- DAUERT ➔ Mit IotTemplate beginnen
- Pfade setzen
  - echo 'export DOTNET_ROOT=$HOME/.dotnet' >> ~/.bashrc
  - echo 'export PATH=$PATH:$HOME/.dotnet' >> ~/.bashrc
  - source ~/.bashrc
- dotnet --version

Erster Test: Simple WebApi

- mkdir dotnet
- cd dotnet
- dotnet new webapi –n FirstTest
- cd FirstTest
- dotnet build
- dotnet run --urls http://0.0.0.0:5000

Ergebnis


Nicht sicher
10.0.0.26:5000/swagger/index.html
Swagger
Select a definition
FirstTest v1
FirstTest
1.0
OAS3
http://10.0.0.26:5000/swagger/v1/swagger.json
WeatherForecast
GET
/WeatherForecast
Parameters
Cancel
No parameters
Execute
Clear
Responses
Curl
curl -X 'GET' \
'http://10.0.0.26:5000/WeatherForecast' \
-H 'accept: text/plain'
Request URL
http://10.0.0.26:5000/WeatherForecast
Server response
Code
Details
200
Response body
[
{
"date": "2022-03-08T12:17:43.8996392+01:00",
"temperatureC": 0,
"temperatureF": 32,
"summary": "Sweltering"
},
{
"date": "2022-03-09T12:17:43.9231045+01:00",
"temperatureC": 18,
"temperatureF": 64,
"summary": "Balmy"

Infrastruktur läuft