

Raspberry Pi auf Sd-Card installieren

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

Downloads

Raspberry Pi OS (previously called Raspbian) is our official operating system for **all** models of the Raspberry Pi.

Use **Raspberry Pi Imager** for an easy way to install Raspberry Pi OS and other operating systems to an SD card ready to use with your Raspberry Pi:

- [Raspberry Pi Imager for Windows](#)
- [Raspberry Pi Imager for macOS](#)
- [Raspberry Pi Imager for Ubuntu](#)

Version: 1.4

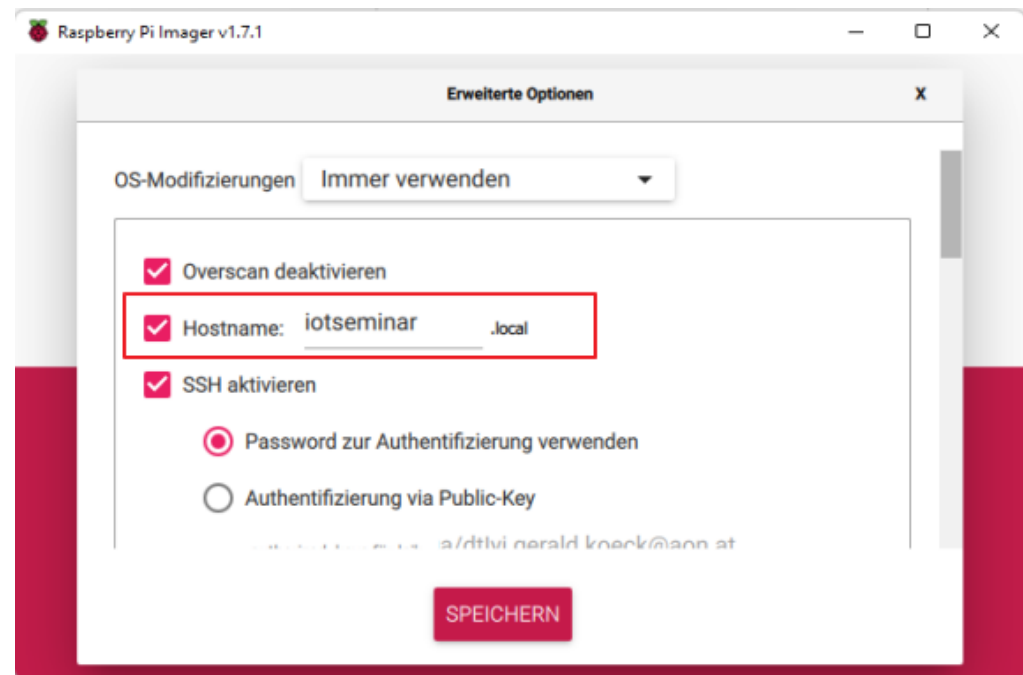
Imager anwenden

- 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

Benutzer > geral > .ssh

Name	Änderungsdatum	Typ	Größe
 id_rsa	12.10.2016 18:07	Datei	2 KB
 id_rsa.pub	12.10.2016 18:07	Microsoft Publish...	1 KB
 id_rsa	13.10.2017 12:41	Datei	1 KB
 known_hosts	04.12.2019 10:04	Datei	7 KB

known_hosts

```
33 192.168.0.89 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGRj1QeSHMNSj2lYnN/x6MkyKFjuwrb7ud1niGvaPvxw0b
34 192.168.0.108 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDu2mt0lhq5KbbJQYVQgH3WLNhofFdmUB9OH0IRFSreA1
35 192.168.0.105 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKFtIEuNkj5HUrIg9gS1cLJ1UDVkkmDZA28YAVT9kzJA
36 192.168.0.56 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKFtIEuNkj5HUrIg9gS1cLJ1UDVkkmDZA28YAVT9kzJAM
37 192.168.0.58 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBBjPE42ur2+x2cudXUxsDsASuUpXG6zH1DoLaU0sv3Fbzb
38
```

Wird gleich wieder eingetragen

```
PS D:\work\IOT\IotSamstag> ssh pi@192.168.0.105  
The authenticity of host '192.168.0.105 (192.168.0.105)' can't be established.  
ECDSA key fingerprint is SHA256:EcdcRitiAenorXjWMqZGgjfoyz0YvtwsxRD2Xk3oCNw.  
Are you sure you want to continue connecting (yes/no)? yes
```

- Einstiegsaccount pi/raspberry

Bootloader bei Bedarf aktualisieren

```
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

FRITZ!Box 7590

Heimnetz > Netzwerk

Netzwerkverbindungen Netzwerkeinstellungen

 Homematic	 LAN 2 mit 1 Gbit/s	192.168.0.59
 raspberrypi	 LAN 4 mit 100 Mbit/s	192.168.0.89

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`

```
GNU nano 3.2 /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=AT

network={
    ssid="linksysAtHome"
    psk="1234567890"
}

network={
    ssid="A1-B5035B"
    psk="1234567890"
}
```

Samba installieren

- <https://www.elektronik-kompodium.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]

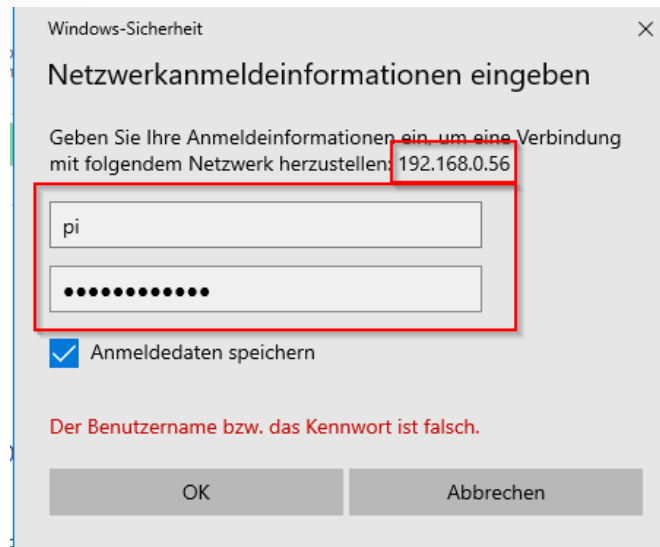
comment = opt
path = /opt/
writeable = yes
browsable = yes

[pi]

comment = home pi
path = /home/pi/
writeable = yes
browsable = yes

- Restart
 - sudo service smbd restart

Samba testen



192.168.0.56 > pi >

Name	Änderungsdatum	Typ	Größe
.gnupg	13.08.2020 12:14	Dateiordner	
.bash_history	13.08.2020 12:31	BASH_HISTORY-D...	1 KB
.bash_logout	27.05.2020 09:10	Bash Logout-Quel...	1 KB
.bashrc	27.05.2020 09:10	Bash RC-Quelldatei	4 KB
.profile	27.05.2020 09:10	PROFILE-Datei	1 KB

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
1  version: "3.8"
2  services:
3    portainer:
4      image: portainer/portainer:latest
5      container_name: portainer
6      command: -H unix:///var/run/docker.sock
7      restart: always
8      ports:
9        - 9000:9000
10       - 8000:8000
11     volumes:
12       - /var/run/docker.sock:/var/run/docker.sock
13       - portainer-data:/data
```

compose-Files strukturieren

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


 docker-compose.yml	07.03.2022 10:57	Yaml-Quelldatei	2 KB
 mosquitto	07.03.2022 11:30	Dateiordner	
 dynv6-updater	07.03.2022 11:30	Dateiordner	

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

Portainer läuft

- 192.168.0.56:9000

192.168.0.89:9000/#/init/admin



Please create the initial administrator user.

Username


Password

Confirm password ☒

✓ The password must be at least 8 characters long

[Create user](#)

192.168.0.89:9000/#/containers



Home PRIMARY

- Dashboard
- App Templates
- Stacks
- Containers**
- Images
- Networks
- Volumes

Container list Containers

Portainer support admin
[my account](#) [log out](#)

Columns Settings

Start Stop Kill Restart Pause Resume Remove Add container

Search...

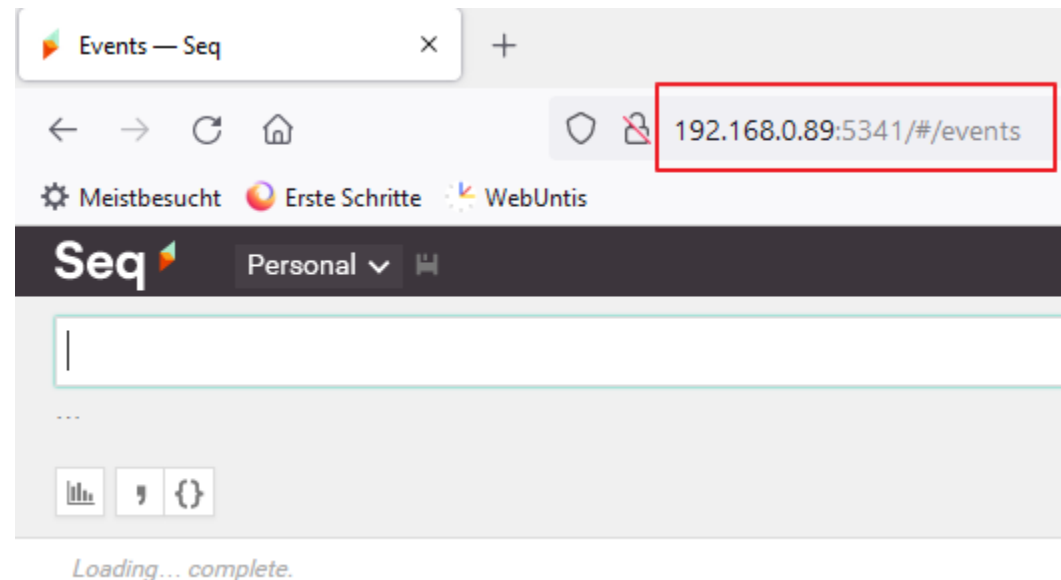
Name	State Filter	Quick actions	Stack	Image	Created	Published Ports	Ownership
<input type="checkbox"/> portainer_portainer_1	running	info logs kill	portainer	portainer/portainer:latest	2020-08-12 12:53:12	8000:8000 9000:9000	administrators

Items per page 10

LogServer Seq

```
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
```

Password erzeugen


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

Container console
[Containers](#) > [mosquitto](#) > Console

>_ Execute

Command 

Use custom command ☐

User 

Connect

Commandline im Container

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

```
/ # cd /mosquitto/config/  
/mosquitto/config # ls -l  
total 8  
-rwxrwxrwx    1 mosquitt mosquitt    298 Aug 18 15:05 mosquitto.conf  
-rwxrwxrwx    1 mosquitt mosquitt    116 Jan  5  2020 passwd  
/mosquitto/config # mosquitto_passwd -b passwd sieglinde [REDACTED]
```

- Kennwort löschen
 - `mosquitto_passwd -D passwordfile user`

Kontrolle im File

```
/mosquitto/config # cat passwd
gerald:$6$jK36WkI+kiIjx8zm$N7pBKLK/0CjCaH7+TEe71tQwtqWEWQmPI86Sm9qoGZtxptIjDRqfxKCavtjz25ZtRPXv80n0po3pnKepB6jfBA==
sieqlinde:$6$K7bBDoGTUV+vqaBw$/EqFtL1EG37D59NI7VAePt1fxBpt7ELvJ90QiXXt6pvknPS1DIzWPDj7VoFucvQJK6MBd0dH1+U0eGZMrfJxKw==
/mosquitto/config #
```

Kontrolle des externen Zugriffs

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

MQTT Connection mqtt://10.0.0.26:8883/

Name

☐ Validate certificate

☒ Encryption (tls)

Protocol

Host

Port

Username

Password

Zugriff auf Logfile über Volume

192.168.0.66 > pi > docker > volumes > mosquitto > log

Name	Änderungsdatum	Typ	Größe
 mosquitto.log	18.08.2020 15:33	Textdokument	5 KB

≡ mosquitto.log ✕

≡ 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



Version

Processor: **BCM2835**
Distribution: **Debian GNU/Linux 11 (bullseye)**
Kernel version: **Linux 5.10.92-v8+ aarch64**
Firmware: **#1514**
Package(s): **0 upgradable(s)**

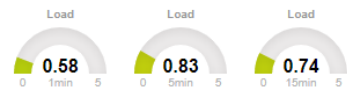


Uptime

Raspberry Pi time: **19:36:43**
Uptime: **38 minutes 49 seconds**



CPU

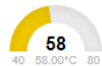


CPU frequency: **600MHz** Voltage: **undefinedV**
Scaling governor: **ondemand**



Temperature

CPU Temperature



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

The screenshot displays the Swagger UI for a service named "FirstTest" (version 1.0, OAS3). The selected definition is "FirstTest v1". The endpoint being viewed is "GET /WeatherForecast".

Parameters: No parameters are defined for this endpoint.

Execute: A blue button to execute the request. A "Clear" button is also present.

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:** 200
- Details:**
- 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": 10,
  "temperatureF": 64,
  "summary": "Balmy"
}
```



Infrastruktur läuft

Container list

Containers

 Containers

 Start  Stop  Kill  Restart  Pause  Resume  Remove  Add container

 Search...

<input type="checkbox"/> Name	State  Filter 	Quick actions	Stack	Image	Created	Published Ports
<input type="checkbox"/> rpi-monitor	running	   	docker	michaelmiklis/rpi-monitor:latest	2022-03-07 12:01:37	 8888:8888  8888:8888
<input type="checkbox"/> mosquito	running	   	docker	eclipse-mosquitto	2022-03-07 11:51:15	 1883:1883  1883:1883  8883:8883  8883:8883  9001:9001  9001:9001
<input type="checkbox"/> seq	running	   	docker	datahust/seq:2021.3.6681-arm64	2022-03-07 11:47:36	 5341:80  5341:80
<input type="checkbox"/> portainer	running	   	docker	portainer/portainer:latest	2022-03-07 11:35:50	 8000:8000  8000:8000  9000:9000  9000:9000