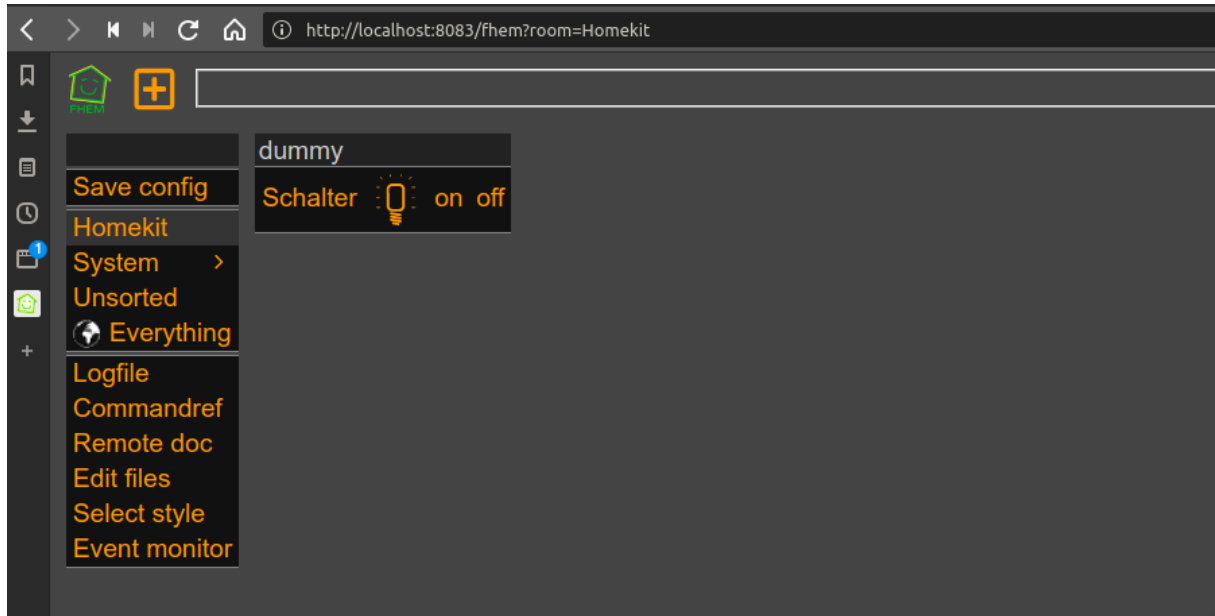

Home Automation Stack



The stack contains everything to run FHEM on a Docker host. Mosquitto is used as message broker. SIRI functions are realized with the help of a homebridge container. The complete stack runs on x86 as well as arm architectures. It is very easy to clone its complete productive environment and has a simple way to build a test system.

Todo

- deCONZ Image Container Integration
- DBLog Integration

Requirements

- docker
- docker-compose

Installation raspberrypi

System Update

- ```
1 sudo apt-get update
2 sudo apt-get upgrade
```

---

### Raspberry Config

```
1 sudo raspi-config
2 sudo reboot
```

### Intall additional packages

```
1 sudo apt-get install wget git apt-transport-https vim telnet
```

### Install docker

```
1 curl -sSL https://get.docker.com | sh
2 sudo systemctl enable docker
3 sudo systemctl start docker
4 sudo usermod -aG docker pi
```

### git repository export

```
1 cd
2 git clone https://github.com/stormmurdoc/fhemdocker.git
3 cd fhmdocker
```

### Installation docker compose

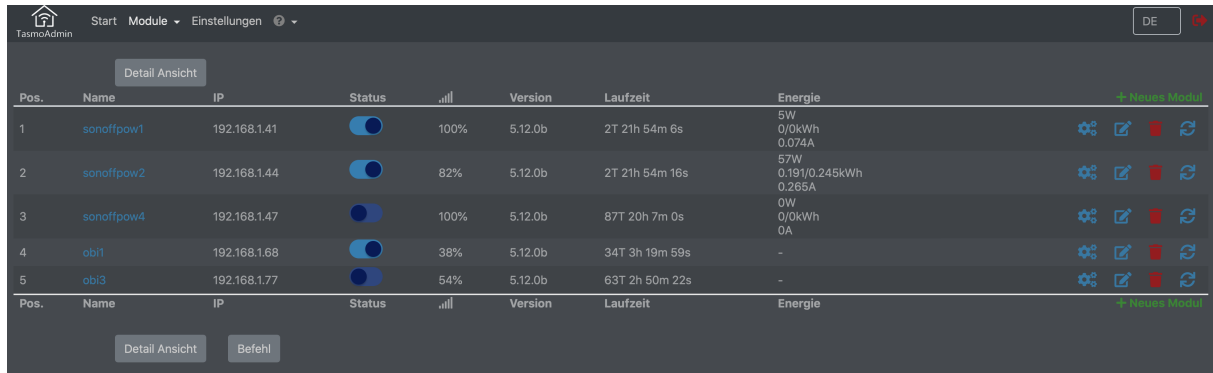
```
1 sudo apt-get install python-pip
2 sudo pip install docker-compose
```

### Start all container

```
1 docker-compose up
```

## Container

### Tasmota Admin

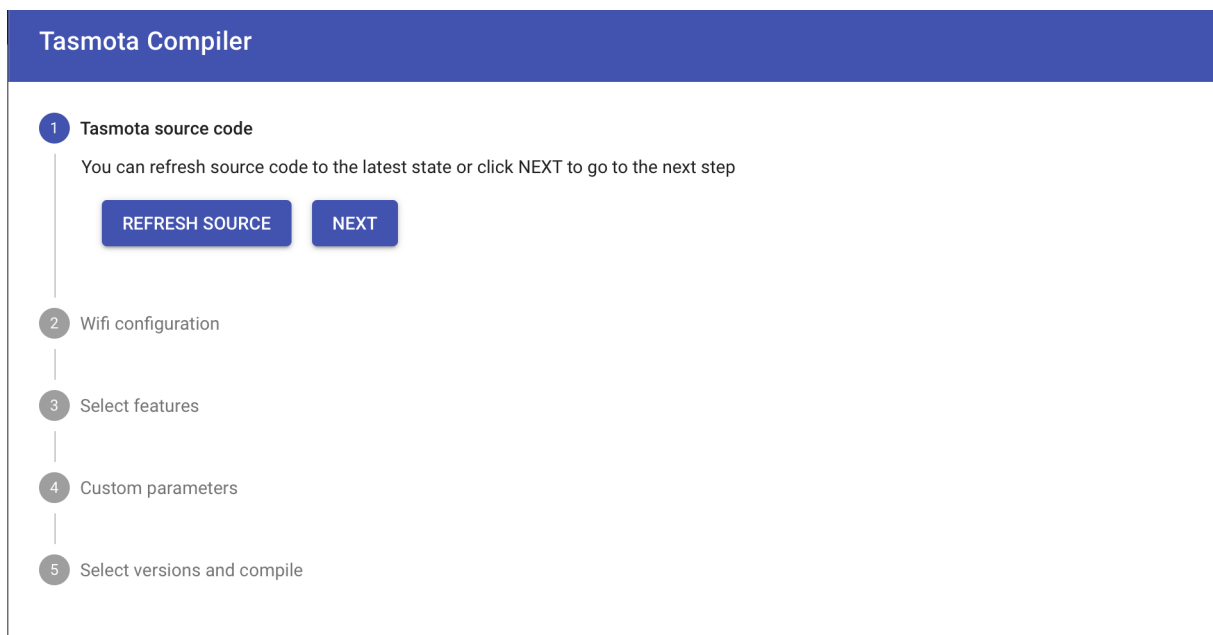


The screenshot shows the Tasmota Admin web interface. At the top, there is a navigation bar with 'Start', 'Module', and 'Einstellungen'. A 'DE' button is in the top right. Below the navigation bar, there is a 'Detail Ansicht' button. The main content is a table with columns: Pos., Name, IP, Status, Signal strength, Version, Laufzeit, and Energie. There are five rows of modules listed. Each row has a set of icons (gear, edit, stop, refresh) on the right. Below the table, there are 'Detail Ansicht' and 'Befehl' buttons.

| Pos. | Name       | IP           | Status | Signal | Version | Laufzeit       | Energie                         | Icons      |
|------|------------|--------------|--------|--------|---------|----------------|---------------------------------|------------|
| 1    | sonoffpow1 | 192.168.1.41 | On     | 100%   | 5.12.0b | 2T 21h 54m 6s  | 5W<br>0/0kWh<br>0.074A          | ⚙️ ✎️ ⏏️ ↺ |
| 2    | sonoffpow2 | 192.168.1.44 | On     | 82%    | 5.12.0b | 2T 21h 54m 16s | 57W<br>0.191/0.245kWh<br>0.265A | ⚙️ ✎️ ⏏️ ↺ |
| 3    | sonoffpow4 | 192.168.1.47 | On     | 100%   | 5.12.0b | 87T 20h 7m 0s  | 0W<br>0/0kWh<br>0A              | ⚙️ ✎️ ⏏️ ↺ |
| 4    | obi1       | 192.168.1.68 | On     | 38%    | 5.12.0b | 34T 3h 19m 59s | -                               | ⚙️ ✎️ ⏏️ ↺ |
| 5    | obi3       | 192.168.1.77 | On     | 54%    | 5.12.0b | 63T 2h 50m 22s | -                               | ⚙️ ✎️ ⏏️ ↺ |

Abbildung 1: “tasmotaadmin”

### Tasmota Compiler



The screenshot shows the Tasmota Compiler web interface. It has a blue header with the title 'Tasmota Compiler'. Below the header, there is a vertical list of steps: 1. Tasmota source code, 2. Wifi configuration, 3. Select features, 4. Custom parameters, and 5. Select versions and compile. Step 1 is currently active. Below step 1, there is a text prompt: 'You can refresh source code to the latest state or click NEXT to go to the next step'. Below this text are two buttons: 'REFRESH SOURCE' and 'NEXT'.

**Tasmota Compiler**

- 1 Tasmota source code**  
You can refresh source code to the latest state or click NEXT to go to the next step  
[REFRESH SOURCE](#) [NEXT](#)
- 2 Wifi configuration
- 3 Select features
- 4 Custom parameters
- 5 Select versions and compile

Abbildung 2: “tasmotacompiler”

## Homebridge

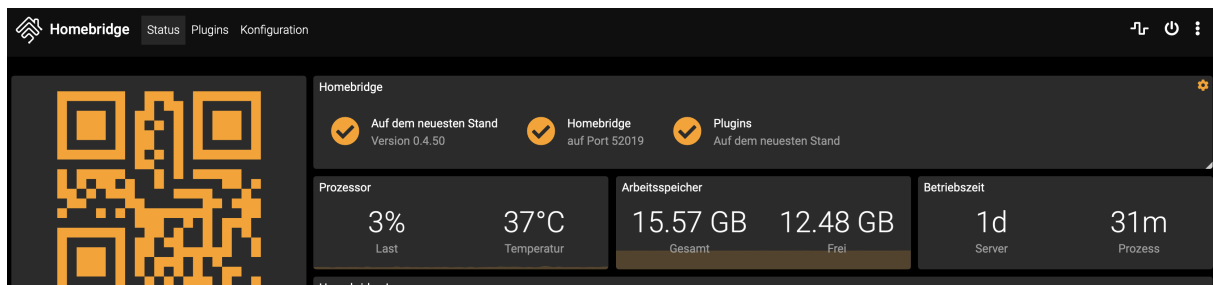


Abbildung 3: “homebridge”

## Portainer

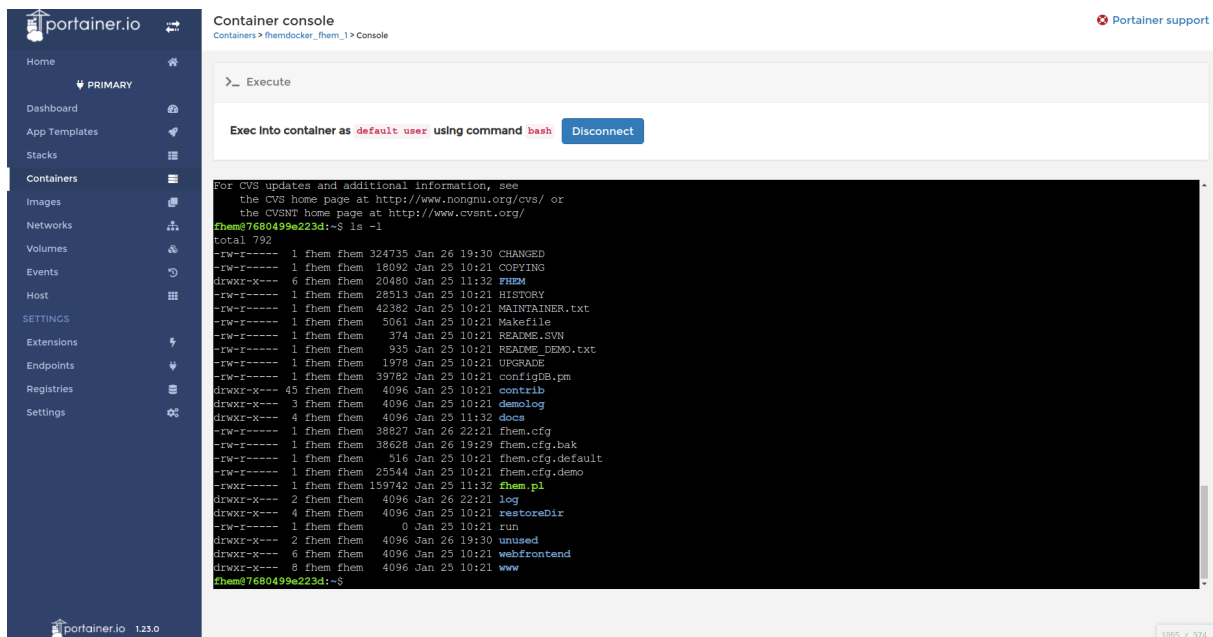


Abbildung 4: “portainer”

## ctop

### Description

ctop is a commandline monitoring tool for linux containers

---

```
ctop - 11:34:11 CET 7 containers
```

| NAME                        | CID          | CPU | MEM           | NET RX/TX | IO R/W    | PIDS |
|-----------------------------|--------------|-----|---------------|-----------|-----------|------|
| • fhemdocker_adminer_1      | e44706321c50 | 0%  | 4M / 15.57G   | 7K / 220B | 0B / 0B   | 1    |
| + • fhemdocker_fhem_1       | fa2b2468390c | 5%  | 77M / 15.57G  | 18K / 39K | 0B / 276K | 4    |
| • fhemdocker_homebridge_1   | b914d7962583 | 0%  | 128M / 15.57G | 0B / 0B   | 0B / 60K  | 39   |
| • fhemdocker_portainer_1    | 857f74666414 | 0%  | 9M / 15.57G   | 15K / 1K  | 0B / 148K | 18   |
| • fhemdocker_tasmocompil... | ffdfed452715 | 0%  | 41M / 15.57G  | 7K / 110B | 0B / 4K   | 18   |
| • fhemdocker_tasmotaadmi... | fc63ff98efe7 | 0%  | 12M / 15.57G  | 8K / 110B | 0B / 8K   | 10   |
| • fhemdocker_broker_1       | c442b745308f | -   | -             | -         | -         | -    |

**Abbildung 5:** “ctop”

## Installation

ctop is available in AUR, so you can install it using AUR helpers, such as YaY, in Arch Linux and its variants such as Antergos and Manjaro Linux.

### Installation Linux

```
1 sudo wget https://github.com/bcicen/ctop/releases/download/v0.7.3/
 ctop-0.7.3-linux-amd64 -O /usr/local/bin/ctop
2 sudo chmod +x /usr/local/bin/ctop
```