

Server - Installation mit Docker / Podman

Neben der Installation durch [Selbstkompilieren](#) oder [Installieren mit vorkonfigurierten Setups](#) kann BFFH relativ simpel als Container appliziert werden. Wir beschreiben hier exemplarisch die Installation mit Docker. Es gibt außerdem alternative Container-Dienste wie z.B. Podman. Wir freuen uns hier über Zuarbeit in der Dokumentation.

Intro

Es gibt prinzipiell die Wege mit Docker Compose (hierzu erstellen wir eine `docker-compose.yml` Datei) oder nutzen den simplen `docker run` Befehl.

Das in beiden Fällen dafür nötige Docker-Image können wir dabei entweder aus dem Repository per `Dockerfile` selbst erzeugen oder aus der [BFFH Container Registry](#) laden (pull).

Außerdem gibt es verschiedene Testumgebungen zum Ausprobieren, welche mehr oder weniger gepflegt sind. Diese enthalten `docker-compose.yml` Dateien.

Container bieten eine relativ generische und simple Methode, um mehrere Instanzen parallel zu betreiben, schnell zu modifizieren und wieder zu zerstören. Wir können damit relativ sauber mehrere Instanzen voneinander isolieren - zum Beispiel mehrere Produktivinstanzen oder ein Testsystem von einem Stagingsystem. Container bieten durch aus viele Möglichkeiten und Vorteile. Klare Nachteile sind der erhöhte Administrations- und Ressourcenaufwand, sowie zusätzliche Konfiguration von Firewalls, Upgrades, Deploy-Scripts oder das kompliziertere Einbinden von externen Tools - wieim Falle von BFFH beispielsweise das Einbinden von Monitoring oder Aktoren.

Anforderungen und Kompatibilität

Mindestanforderungen und kompatible Hardware finden sich [hier](#).

Installation mit Docker

Diese Beschreibung ist für ein Ubuntu-System erstellt und ist analog anzuwenden/anzupassen für andere Systeme wie Fedora, CentOS, Arch Linux, etc.

Docker installieren

Siehe auch <https://phoenixnap.com/kb/docker-on-raspberry-pi>

```
sudo apt update && sudo apt upgrade
sudo apt install ca-certificates curl gnupg lsb-release

sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
```

bffh Benutzer anlegen

Ein gesonderter Nutzer wird den Container als Non-Root ausführen. Es ist ein Systembenutzer ohne eigenes Homeverzeichnis (`-s` Parameter).

```
sudo useradd -m -s /bin/bash bffh
```

Docker Image erstellen mit Dockerfile

TODO

Docker Compose installieren

Siehe auch <https://dev.to/elalemany0/how-to-install-docker-and-docker-compose-on-raspberry-pi-1mo>

```
sudo useradd -m -s /bin/bash bffh
```

BFFH Git Projekt klonen

Get Docker-Compose Files `git clone`

`https://gitlab.com/fabinfra/fabaccess/dockercompose.git fabaccess-server`

The Dockerfile is in the root directory of the main repo `docker-compose.yml` is available in a separate [git repo](#)

Edit config files in `config` folder to taste

Start Server with `docker-compose up -d`

To make it easier to apply your changes in your config and keep the dockercompose up to date, you should “fork” this repository.

Server Logs

Wir können auch die Logs einsehen, um zu prüfen, ob alles funktioniert:

```
docker-compose logs
```

Docker mit [Certbot](#) oder [DNSRoboCert](#)

Mit Hilfe von Certbot oder DNSRoboCert können wir automatisch und einfach SSL-Zertifikate für BFFH und für Mosquitto erstellen.

ToDo

Version #9

Erstellt: 2024-10-23 22:57:17 CEST von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 2025-03-11 20:53:31 CET von Mario Voigt (Stadtfabrikanten e.V.)