

Server - Anforderungen und Kompatibilität

BFFH kann auf einer ganzen Reihe von Systemen zu Laufen gebracht werden.

Kompatible Architekturen

Wir bieten die fertige `bffhd` und `fabfire_provision` binaries in GitLab und im [FabAccess Server Paket \(*.deb, *.rpm\)](#) für folgende Architekturen an:

Rust Target	Debian	Auch bekannt als	Adressierung
<code>x86_64-unknown-linux-gnu</code>	amd64	x86-64, x86_64, AMD64 oder x64	64 Bit
<code>aarch64-unknown-linux-gnu</code>	arm64	ARMv8	64 Bit
<code>armv7-unknown-linux-gnueabi</code>	armhf	ARMv7	32 Bit

Konkrete Beispielhardware: kompatibel sind beispielweise Raspberry Pi Zero 2 W, Pi 2, Pi 3, Pi 4 und Pi 5, sowie diverse Synology NAS und natürliche alle gewöhnlichen Computer und Server. Nicht kompatibel sind jedoch Raspberry Pi 1, Raspberry Pi Zero und Raspberry Pi Zero 2. Die nachfolgende Tabelle zeigt ein paar weitere Details.

Übersicht gängiger Geräte (Raspberry Pi, Synology) und deren Architekturen (grün = unterstützt, orange = nicht unterstützt).

Modell	CPU	Architektur	Erschienen	Rust Compiler Target	Kommentar
Raspberry Pi Zero	ARM11	ARMv6 (32 Bit)	11/2015	<code>arm-unknown-linux-gnueabi</code> <code>arm-unknown-linux-gnueabi</code>	Betrachten wir als deprecated, weil über 10 Jahre alt und weil 32 Bit

Raspberry Pi Zero W	ARM11	ARMv6 (32 Bit)	02/2017	arm-unknown-linux-gnueabi arm-unknown-linux-gnueabihf	Betrachten wir als deprecated, weil über 10 Jahre alt und weil 32 Bit
Raspberry Pi Zero 2 W	ARM Cortex-A	ARMv8 (64 Bit)	10/2021	aarch64-unknown-linux-gnu	
Raspberry Pi 1 Model A	ARM1176JZF-S	ARMv6 (32 Bit)	02/2013	arm-unknown-linux-gnueabi arm-unknown-linux-gnueabihf	Betrachten wir als deprecated, weil über 10 Jahre alt und weil 32 Bit
Raspberry Pi 1 Model A+	ARM1176JZF-S	ARMv6 (32 Bit)	11/2014	arm-unknown-linux-gnueabi arm-unknown-linux-gnueabihf	Betrachten wir als deprecated, weil über 10 Jahre alt und weil 32 Bit
Raspberry Pi 1 Model B+	ARM1176JZF-S	ARMv6 (32 Bit)	04/2012	arm-unknown-linux-gnueabi arm-unknown-linux-gnueabihf	Betrachten wir als deprecated, weil über 10 Jahre alt und weil 32 Bit
Raspberry Pi 1 Model B+	ARM1176JZF-S	ARMv6 (32 Bit)	07/2014	arm-unknown-linux-gnueabi arm-unknown-linux-gnueabihf	Betrachten wir als deprecated, weil über 10 Jahre alt und weil 32 Bit
Raspberry Pi 2 Model B	Cortex-A7	ARMv7 (32 Bit)	02/2015	armv7-unknown-linux-gnueabihf	
Raspberry Pi 2 Model B v1.2	Cortex-A53	ARMv8 (64 Bit)	09/2016	aarch64-unknown-linux-gnu	
Raspberry Pi 3 Model A+	Cortex-A53	ARMv8 (64 Bit)	11/2018	aarch64-unknown-linux-gnu	

Raspberry Pi 3 Model B	Cortex-A53	ARMv8 (64 Bit)	02/2016	aarch64-unknown-linux-gnu	
Raspberry Pi 3 Model B+	Cortex-A53	ARMv8 (64 Bit)	03/2018	aarch64-unknown-linux-gnu	
Raspberry Pi 4 Model B	Cortex-A72	ARMv8 (64 Bit)	07/2019	aarch64-unknown-linux-gnu	
Raspberry Pi 5	Cortex-A76	ARMv8 (64 Bit)	10/2023	aarch64-unknown-linux-gnu	
Synology Ds218j	Armada38x	ARMv7 (32 Bit)	10/2017	armv7-unknown-linux-gnueabi	
Synology RS819	rtd1296	ARMv8 (64 Bit)	04/2019	aarch64-unknown-linux-gnu	
Synology DS718+	Intel Celeron J3455	amd64 (64 Bit)	09/2017	x86_64-unknown-linux-gnu	
A20-OLinuXino-LIME2	Allwinner A20/T2 (Cortex-A7)	ARMv7 (32 Bit)	03/2015	armv7-unknown-linux-gnueabi	

Architekturreferenzen

- Raspberry Pi
 - https://de.wikipedia.org/wiki/Raspberry_Pi#Raspberry-Pi-Modelle
- Synology
 - https://kb.synology.com/de-de/DSM/tutorial/What_kind_of_CPU_does_my_NAS_have
 - <https://github.com/SynoCommunity/spksrc/wiki/Architecture-per-Synology-model#rtd1296-armv8>
- Debian Pakete
 - <https://wiki.debian.org/SupportedArchitectures>
- Rust
 - <https://doc.rust-lang.org/nightly/rustc/platform-support.html>

Kompatible Betriebssysteme

- auf Linux/Unix-Basis, zum Beispiel ...
 - Ubuntu/Kubuntu
 - Arch Linux
 - Raspberry OS
 - Fedora
 - CentOS
 - NixOS → Luca Lutz vom Hackwerk e.V. fragen
 - Synology
 - WSL (Windows Subsystem for Linux)
- FreeBSD
- MacOS (**aktuell ungetestet!**)

Hinweis: BFFH kann **nicht** auf Windows nativ kompiliert werden. Das liegt an der `winapi`. Dazu gibt es auch einen [Kommentar](#).

Geeignete Container/Virtualisierungswerkzeuge

Wer BFFH nicht nativ installieren möchte, kann auch eine Containerumgebung [und entsprechend damit aufsetzen](#), zum Beispiel:

- [Docker](#)
- [Portainer](#)
- [Kubernetes](#) → Luca Lutz vom Hackwerk e.V. fragen
- [Moby](#)
- [Podman](#)
- [Proxmox](#)
- [LXC](#)
- [runC](#)
- [Containerd](#)
- [VirtualBox](#)
- [Boxes](#)
- ...

Empfehlungen für Hardware

Allgemeine Empfehlungen sind immer relativ schwer zu treffen. Je nach Größe der Institution und der vorhandenen Software- und Personallandschaft gibt es unterschiedlichste Auffassungen davon, was benötigt wird und wie es mit anderen Systemkomponenten zusammenspielen soll bzw. muss. Grundsätzlich versuchen wir, FabAccess Server mit möglichst wenig Ressourcenverbrauch zu installieren. Grundsätzlich benötigt:

- **Festplatte:**

- BFFH benötigt prinzipiell kaum Speicherplatz. Die Binary ist ca. 200 MB groß. Dazu kommen noch winzige Konfigurationsdateien. Allerdings sollten Auslagerungsdatei, Systemdateien und Platz für Umgebungen wie python3-venv gelassen werden.
- BFFH schreibt u.U. fleißig Log-Files (Audit). Außerdem werden ggf. weitere Systeme installiert, wie z.B. Scripts oder Monitoring-Services.
- Wir empfehlen deshalb pauschal 16 GB oder mehr (für Betriebssystem, BFFH, Services, Log Files und Sonstige + Puffer)
- idealerweise eine schnelle SSD oder Industrial Grade SD-Karte (mit erhöhter Lebensdauer)

- **Arbeitsspeicher:** $\geq 1,5$ GB (nur zum Kompilieren der Binary - trifft nicht für die fertig ausführbare `bffhd` Binary zu!)

- **CPU Kerne:** ≥ 1

- **Netzwerkkarte** (idealerweise LAN, nicht nur Wifi)

Sonstige Empfehlungen in der Werkstatt

- u.U. sinnvoll: USV (unterbrechungsfreie Stromversorgung)
- **stabiles** Wifi-Netzwerk (viele Akteure werden u.U. lediglich per WLAN angebunden!)

Weiterführende Informationen zu Empfehlungen sind im Guide [Getting Started / Onboarding](#) beschrieben.

Version #13

Erstellt: 20 Februar 2025 15:59:19 von Mario Voigt (Stadtfabrikanten e.V.)

Zuletzt aktualisiert: 15 März 2025 13:03:52 von Mario Voigt (Stadtfabrikanten e.V.)