Server Dokumentation

- Namensschema
- Hosts
  - asgard
  - bifroest
  - heimdal
  - hermod
  - thor
  - valhalla

Namensschema

Als Namensschema für die Hostnamen der **Server** im Maschinendeck wird "*Nordische Mythologie*" gewählt.

Hosts

- `asgard`
- `thor`
- `valhalla`

Hosts

asgard

Der Host **`asgard`** ist der interne Haupserver im deckeigenen Rechenzentrum.

IP: 192.168.0.107

Hostnamen:

- `asgard.maschinendeck.org`
- `tuer.maschinendeck.org`
- `docs.maschinendeck.org`
- `nextcloud.maschinendeck.org`
- `api.maschinendeck.org`
- `nextcloud.maschinendeck.org`
- `social.maschinendeck.org`

Spezifikation

CPU	Intel Celeron J4105 CPU, *4 x 1.50GHz*
GPU	Intel UHD Graphics 600
RAM	8 Gb, DDR4, *2400 MT/s*
Speicher	1 Tb, *Intenso SSD SATAIII*
Netzwerk	1000BASE-T
OS	Ubuntu 24.04.1 LTS

Aufbau

Der Server läuft vollständig im Dockermodus. Die Konfiguration der Container liegt unter `/config`, deren Daten unter `/data` und etwaige Backups der Datenbanken und der in den Containern laufenden Dienste unter `/backup`.

Die Umgebungsvariablen für einen Container, welche unter Umständen auch Passwörter oder Zugriffstoken enthalten, befinden sich in einer Datei mit dem Namen `.env` in dem jeweiligen Verzeichnis unter `/config`.

Stack

Ankommende `HTTP`-Anfragen werden von einem traefik Reverseproxy entgegengenommen, welcher diese an die unterliegenden Dockercontainer weiterreicht. Der `traefik`-Container kümmert sich vollautomatisch um die SSL-Zertifikate via *Letsencrypt*.

Zur Konfiguration von `traefik` für einen einzelnen Container, muss die `docker-compose.yml` des Stacks lediglich mit entsprechenden Labels ausgestattet werden. Das Routing auf den exponierten Port geschieht dann automatisch. Hier eine beispielhafte Konfiguration für ein Standardsetup:

```
services:
  example:
    image: example/example
    container_name: example
    env_file: .env
    expose:
      - 80
    networks:
      - public
      - mysql
    labels:
      - traefik.enable=true
      - traefik.http.routers.example.rule=Host(`example.maschinendeck.org`)
      - traefik.http.routers.example.entrypoints=web
      - traefik.http.routers.example_ssl.rule=Host(`example.maschinendeck.org`)
      - traefik.http.routers.example_ssl.tls=true
      - traefik.http.routers.example_ssl.tls.certresolver=letsencrypt
      - traefik.http.routers.example_ssl.entrypoints=websecure
    restart: unless-stopped

networks:
  public:
    external: true
```

Der entsprechende Container muss mindestens dieses Set an Labels aufweisen um funktionieren zu können *(die Strings `example` und `example_ssl` sind für jeden Stack einzigartig zu wählen, müssen also geändert werden)*. Zusätzlich muss er Zugriff auf das Netzwerk `public` erhalten und selbstverständlich muss die entsprechende Domain auf den *DynDNS* des Maschinendeck aufgeschaltet sein.

Externe Dienste

Auch alle Anfragen an interne Dienste, welche nicht auf dem Server, sondern auf anderen Hosts im internen Netzwerk betrieben werden, landen bei diesem `traefik`-Container. Die Konfiguration für die Handhabung solcher Dienste befindet sich in der Datei `/config/traefik/config/dynamic.yaml`, wo entsprechende Services und Router hinterlegt sind.

Da dort Anfragen in der Regel stupide an einen anderen Host weitergeleitet werden, bestehen diese Konfigurationen aus simplen `loadBalancer`-Direktiven:

```
http:
  services:
    tuer_service:
      loadBalancer:
        servers:
          - url: http://192.168.0.101
  routers:
    tuer:
      service: tuer_service
      entryPoints:
        - web
        - websecure
      rule: Host(`tuer.maschinendeck.org`)
      tls:
        certResolver: letsencrypt
```

Jeweils ein Service und ein Router müssen pro externem Dienst angelegt werden.

Aktuelle Dienste

Zur Zeit stellt der Host `asgard` folgende Dienste zur Verfügung:

bookstack

Die Software bookstack stellt die Dokumentation zur Verfügung, die hier gerade zu sehen ist. Jeder der eine @maschinendeck.org E-Mail-Adresse besitzt kann sich hier ein Konto erstellen und an der Dokumentation mitwirken.

Mastodon

Diese Mastodon-Instanz ist unser Standbein im Fediverse. Alle Maschinisten können einen Account auf Einladung erhalten.

Nextcloud

Bei Nextcloud handelt es sich um eine Open Source "private cloud"-Lösung, auf welcher Maschinisten Dateien ablegen und Kalender pflegen können.

Bitwarden

Mit der Open Source Impementierung "*vaultwarden*" unterhält das Maschinendeck hier seine eigene Bitwarden-Instanz. Hierbei handelt es sich um einen Passwortmanager, mit dessen Hilfe Passwörter sicher mit ganzen Personengruppen geteilt werden können.

bifroest

Der Host `bifroest` handhabt die Hausautomatisierung/das IoT im Maschinendeck.

IP: 192.168.0.10
IP: 10.23.42.10

Hostnamen:

- `bifroest.local`

Spezifikation

Pi Version	Raspberry Pi 3 Model B
RAM	1 Gb
Netzwerk	300BASE-T
OS	raspberrypiOS bookworm

Aufbau

Der Raspberry Pi bifroest läuft im Dualbetrieb Docker/"bare metal". Die Software Home Assistant *(HASSIO)* läuft in einem Dockercontainer, andere Software *(wie der MQTT-Broker* `mosquitto` *)* ist direkt über das Betriebssystem installiert.

Die Dockerkonfiguration findet sich unter `/opt/docker`.

Peripherie

Der Pi spannt mithilfe des slae.sh-Sticks ein Zigbeenetzwerk auf, in welchem sich zahlreiche Sensoren und Aktoren befinden.

heimdal

Der Host `heimdal` betreibt das **Maschinendeck Door Opening System** *(MDoS)*.

IP: 192.168.0.101

Hostnamen:

- `heimdal.local`
- `tuer.maschinendeck.org`

Spezifikation

Pi Version	Raspberry Pi 3 Model B
RAM	1 Gb
Netzwerk	100BASE-T
OS	raspberrypiOS bullseye

Peripherie

An den Raspberry Pi ist die mdos-Hardware angeschlossen, welche sich direkt am Türschloss befindet. Die Verbindung zum Türmodul wird über eine `RS485` Bustreiber hergestellt, welcher direkt über die Serielle Schnittstelle des Pis angebunden ist. Im Gehäuse mit dem Pi befindet sich ein Leistungsstarkes 12V Netzteil, welches den Pi und das Türgerät mit Strom versorgt.

Aufbau

Der Raspberry Pi wird mit einem *rasperrypiOS lite* betrieben, auf welchem "bare metal" die eigens entwickelte Software mdos-web läuft.

Die Software läuft unter dem Benutzer `pi` und liegt unter `/home/pi/mdos-web`. Gestartet wird Sie über das Node.js-Modul *pm2*, welches dafür sorgt, dass die Anwendung als Daemon im Hintergrund läuft. Die Logs werden in der Datei `/var/log/mdos-web/production` hinterlegt, welche von *logrorate* rotiert wird.

hermod

Der Host `hermod` ist die interne PBX des Maschinendeck.

IP: 192.168.0.XXX

Hostnamen:

- `hermod.local`

Spezifikation

Pi Version	Raspberry Pi 3 Model B
RAM	512 Mb
Netzwerk	1000BASE-T
OS	RasPBX

Aufbau

thor

Der Host `thor` ist ein Server im deckeigenen Rechenzentrum.

IP: 192.168.0.133

Hostnamen:

- `thor.local`

Spezifikation

CPU	AMD G-T48E, 2 *x 700 MHz*
GPU	AMD Radeon HD 6250
RAM	2 Gb, DDR3, 1600 *MT/s*
Speicher	128 Gb, *SK hynix SC300 SATA*
Netzwerk	1000BASE-T
OS	Ubuntu 24.04.1 LTS

valhalla

Der Host `valhalla` ist der interne Backupserver im deckeigenen Rechenzentrum.

IP: 192.168.0.64

Hostnamen:

- `valhalla.local`

Spezifikation

Pi Version	Raspberry Pi 4 Model B
RAM	8 Gb
Netzwerk	1000BASE-T
OS	raspberrypiOS bookworm

Peripherie

Der Raspberry Pi `valhalla` verfügt über **3 Tb HDD** Speicher, welcher via USB verbunden ist. Die Festplatten befinden sich in einem *ICY BOX IB-RD3620SU3* JBOD Gehäuse, welches auf RAID 1 geschaltet ist. Der Grund hierfür ist, dass die genutzte NAS-Software openmediavault keine Software-RAIDs über via USB verbundene Platten erlaubt.

Aufbau

Der Raspberry Pi wird mit einem *rasperrypiOS lite* betrieben, auf welchem "bare metal" die NAS-Software openmediavault *(OMV)* installiert ist. Diese unterteilt die angehangene, externe Festplatte in drei Partitionen *(mangels support in der Weboberfläche wurden diese Partitionen direkt auf dem Betriebssystem mittels der Software `parted` erstellt):*

- `/dev/sda1` → `/share` 100 Gb
- `/dev/sda2` → `/docker` 100 Gb
- `/dev/sda3` → `/backup` 2,8 Tb

Die `/share` Partition stellt ein Netzlaufwerk für den lokalen Dateiaustausch via Samba *(SMB)* zur Verfügung. Dieser Speicher ist als volatil zu betrachten.

Auf der Partition `/docker` werden die compose-Dateien für den von OMV zur Verfügung gestellten Dockerdienst gespeichert. Hier läuft aktuell nur ein einziger Container mit der Software backuppc.

Die Partition `/backup` wird in erwähntem Dockercontainer gemountet, um darauf die Backups der Vereinseigenen IT-Infrastruktur abzulegen.

> **Die Weboberflächen sind ausschließlich via SSL erreichbar, dringend auf `https://` achten.**

openmediavault

Die Weboberfläche der Software openmediavault ist im Deck unter https://valhalla.local erreichbar.

backuppc

Die Weboberfläche der Software backuppc zum Managen und Rückspielen von Backups ist unter https://valhalla.local:8080 erreichbar.