Administration web

Alles die Administration von web betreffend.

Es ist das Paket etckeeper mit Git als Backend installiert um den Vorteil einer Versionsverwaltung für `/etc` zu haben. Nach administrativen Aufgaben einfach kurz

```
sudo etckeeper commit
```

und kurz eine Commit Message eingeben. Das Repository ist ausschließlich lokal vorhanden und wird nicht nach draußen gepusht!

Administratoren

(aka Nutzer mit sudo Rechten)

- tux
- nold
- kwasir

Setup

Netzwerk

Docker-Netzwerk doc0

```
docker network create --driver=bridge --ipv6 --
subnet=fda9:26e:5805:bab1:aeb:d0c0::/96 doc0
```

Installierte Dienste

- Docker
  - Grafana
  - Prometheus
  - MariaDB Backend Wordpress
  - Wordpress
  - InfluxDB
  - Yanic
  - DNS
  - Node Exporter

- Meshviewer Server

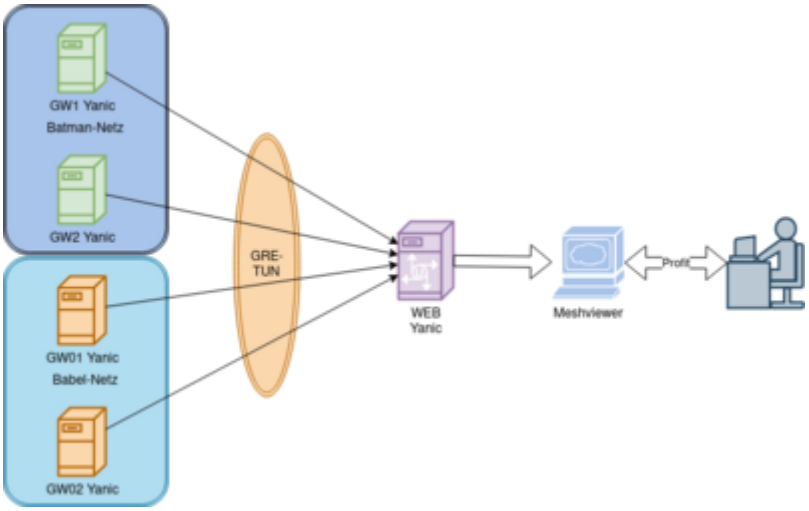
Dienste

Dienste

Query Location	Description
http://web.md.freifunk.net:8010/data/meshviewer.json	Meshviewer nodes.json v2
http://web.md.freifunk.net:8010/#!/de/map	Meshviewer Map

Yanic Konfiguration

Yanic dient der erfassung der Nodes im Batman und Babel Netz Auf jeden Gateways läuft eine Yanic Instanz die Ihre information zur WEB schickt Hier läuft der zentrale Yanic Service der die Daten für den meshviewer bereitstellt.



Der Zentrale Yanic legt seine Daten einmal as JSON für den Meshviewer ab und zusätzich in einer influxDB

Docker Compose 4 all Service

Compose File um alle Services für Freifunk zu starten.

- ListenpunktGrafana
- Prometheus
- InfluxDB
- Yanic
- Meshviewer Server

Location /opt/

File docker-compose.yaml

Config File unter /opt/...

```
version: '3.0'

volumes:
  prometheus-storage:
    external: true
  grafana-storage:
    external: true
networks:
  front-tier:
  back-tier:

services:

  prometheus:
    image: prom/prometheus:latest
    container_name: ffmd_prometheus
    volumes:
      - /opt/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml
      - prometheus-storage:/prometheus
    ports:
      - 9090:9090
    restart: unless-stopped
    networks:
      - back-tier

  grafana:
    image: grafana/grafana:latest
    container_name: ffmd_grafana
    depends_on:
      - prometheus
      - influxdb
    ports:
      - 3000:3000
    environment:
      - GF_INSTALL_PLUGINS:grafana-piechart-panel
    volumes:
      - grafana-storage:/var/lib/grafana
      - /opt/grafana/grafana.ini:/etc/grafana/grafana.ini
    restart: unless-stopped
    networks:
      - front-tier
      - back-tier

  yanic:
    image: ffmd/yanic:latest
    depends_on:
      - influxdb
    network_mode: "host"
    restart: unless-stopped
```

```
    container_name: ffmd_yanic
    volumes:
      - /opt/yanic/config.toml:/etc/yanic.conf

  meshviewer:
    image: ffmd/meshviewer-server:latest
    container_name: ffmd_meshviewer_server
    restart: unless-stopped
    depends_on:
      - yanic
    ports:
      - 8010:80
    environment:
      MeshviewerRepo: "https://github.com/FreifunkMD/Meshviewer --branch
ffmd"
      LoopHookCMD: "wget http://5.252.224.201:8080/data/meshviewer.json -O
/var/www/html/data/meshviewer.json"
    networks:
      - back-tier
      - front-tier
  influxdb:
   environment:
            INFLUXDB_ADMIN_USER: user
            INFLUXDB_ADMIN_PASSWORD: *Ü****
            INFLUXDB_USER: User
            INFLUXDB_USER_PASSWORD: *Ü****
            INFLUXDB_DB: babelstats
            INFLUXDB_READ_USER: Reader
            INFLUXDB_READ_USER_PASSWORD: *Ü****
   image: influxdb:latest
   container_name: ffmd_influxdb
   volumes:
      - /opt/influxdb/data/:/var/lib/influxdb
      - /opt/influxdb/influxdb.conf:/etc/influxdb/influxdb.conf:ro
   restart: unless-stopped
   networks:
     - back-tier
     - front-tier
   ports:
        - "8086:8086"
```

Meshviewer

```
siehe Compose
```

Grafana

Install

```
docker pull grafana/grafana
docker volume create grafana-storage
```

RUN

Siehe compose

Prometheus

Install

```
docker volume create prometheus-storage
docker pull prom/prometheus
```

Config

Start

Sieh Compose

DNS

Der DNS verwaltet die Zone ffmd., für das Docker-Netz. Es ist außerdem eine Weiterleitung auf ns1.netz39.de eingerichtet, der die Zone md.freifunk.net. verwaltet.

Setup des Docker-Containers:

```
docker run -d --restart always --ip6 fda9:26e:5805:bab1:aeb:d0c0::53 --
network doc0 --name bind9-ffmd ffmd/bind9-ffmd
```

- Das Image wird auf Docker Hub verwaltet: ffmd/bind9-ffmd
- Docker-Image auf Github: FreifunkMD/ffmd_bind9
- Konfiguration auf Github: FreifunkMD/ffmd_bind9_cfg_ffmd

Test-Aufruf:

```
dig @fda9:26e:5805:bab1:aeb:d0c0::53 -t AAAA stable-
babel.updates.firmware.ffmd
```

Diese Firewall-Konfiguration mit ip6tables sorgt dafür, dass Aufrufe an fda9:26e:5805:bab1:53::1 an den Docker-Container weitergeleitet werden: (persistiert in `/etc/ufw/before6.rules`)

```
ip6tables -t nat -A PREROUTING  -d fda9:26e:5805:bab1:53::0 -j DNAT --to-
destination fda9:26e:5805:bab1:aeb:d0c0::53
ip6tables -t nat -A OUTPUT -d fda9:26e:5805:bab1:53::0 -j DNAT --to-
destination fda9:26e:5805:bab1:aeb:d0c0::53
ip6tables -A ufw6-before-forward -p udp --dport 53 -d
fda9:26e:5805:bab1:aeb:d0c0::53 -j ACCEPT
ip6tables -A ufw6-before-forward -p tcp --dport 53 -d
fda9:26e:5805:bab1:aeb:d0c0::53 -j ACCEPT
```

Anschließend funktioniert folgender Test-Aufruf:

```
dig @fda9:26e:5805:bab1:53::0 -t AAAA stable-babel.updates.firmware.ffmd
```

Für die Auflösung der Unicast-Adresse müssen Routen angelegt werden, z.B.:

```
ip -6 r a fda9:26e:5805:bab1:53::0 dev backend-web t local
```

Bearbeitet in #13

Node Exporter

Install

```
 curl -s
https://api.github.com/repos/prometheus/node_exporter/releases/latest \
  | grep browser_download_url \
  | grep linux-amd64 \
  | cut -d '"' -f 4 \
  | wget -qi -
 tar xzf node_exporter-0.18.1.linux-amd64.tar.gz
 cd node_exporter-0.18.1.linux-amd64/
 sudo cp node_exporter /usr/local/bin/
```

Config

```
 sudo nano /etc/systemd/system/node_exporter.service

 [Unit]
 Description=Node Exporter
 Wants=network-online.target
 After=network-online.target
 [Service]
 User=prometheus
 ExecStart=/usr/local/bin/node_exporter \
  --collector.cpu \
  --collector.diskstats \
  --collector.filesystem \
```

```
  --collector.loadavg \
  --collector.meminfo \
  --collector.filefd \
  --collector.netdev \
  --collector.stat \
  --collector.netstat \
  --collector.systemd \
  --collector.uname \
  --collector.vmstat \
  --collector.time \
  --collector.mdadm \
  --collector.zfs \
  --collector.tcpstat \
  --collector.bonding \
  --collector.hwmon \
  --collector.arp \
  --web.listen-address=:9100 \
  --web.telemetry-path="/metrics" /etc/systemd/system/node_exporter.service
[Install]
WantedBy=default.target
```

Engage

```
sudo systemctl daemon-reload
sudo systemctl enable node_exporter.service
sudo systemctl start node_exporter.service
```

Wordpress md.freifunk.net

tbd

Docker-Container:

- ffmd_wordpress
- ffmd_mariadb

Migration

Backup Config

Archiv vom gesamten WP Content

```
tar -czf /tmp/website.`date +%Y%m%d-%H%M%S`.tar.gz website
```

Datenbank dump

```
mysqldump --databases website -u website -p > /tmp/wordpress.sql
```

Übertragen der Daten auf den neuen Sever ⇒ Wordpress SQL Dump in das Verzeichnis initdb.d

```
mkdir -pv /var/www/example.com/database/initdb.d
cd /var/www/example.com/database/initdb.d
```

Wordpress Content in das Verzeichnis srv entpacken

```
mkdir -pv /var/www/example.com/src
```

Docker-compose in `/var/www/md.freifunk.net`

docker_compose.yaml

```
version: '3'

networks:
  frontend:
  backend:

services:
  wordpress_db:
    image: mariadb:latest
    container_name: ffmd_mariadb
    volumes:
      - ./database/data:/var/lib/mysql
      - ./database/initdb.d:/docker-entrypoint-initdb.d
    restart: unless-stopped
    networks:
      - backend
    environment:
      MYSQL_ROOT_PASSWORD: **secret**
      MYSQL_DATABASE: **DB_Name**
      MYSQL_USER: **user**
      MYSQL_PASSWORD: **secret**
  wordpress:
    depends_on:
      - wordpress_db
    image: wordpress:latest
    restart: unless-stopped
    container_name: ffmd_wordpress
    networks:
      - backend
      - frontend
    ports:
      - "1234:80"
    environment:
```

```
       WORDPRESS_DB_HOST: wordpress_db:3306
       WORDPRESS_DB_USER: **user**
       WORDPRESS_DB_PASSWORD: **secret**
       WORDPRESS_DB_NAME: **db_namen**
       restart: unless-stopped
    links:
      - wordpress_db:mariadb
    volumes:
      - ./src:/var/www/html
```

starten der

```
docker-compose up -d
```

Apache / Ngingx Config erstellen

---

apache config

```
VirtualHost *:80>
        ServerAdmin kontakt@md.freifunk.net
        ServerName web.md.freifunk.net
        ServerAlias web.md.freifunk.net
        ErrorLog /var/log/apache2/web.md.freifunk.net-error.log
        CustomLog /var/log/apache2/web.md.freifunk.net-access.log common

        Redirect / https://md.freifunk.net/
</VirtualHost>


<VirtualHost *:443>
        ServerAdmin kontakt@md.freifunk.net
        ServerName web.md.freifunk.net
        ServerAlias web.md.freifunk.net
        ErrorLog /var/log/apache2/web.md.freifunk.net-error.log
        CustomLog /var/log/apache2/web.md.freifunk.net-access.log common

        SSLEngine on
        SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
        SSLCertificateFile **
        SSLCertificateKeyFile **
        SSLCertificateChainFile **


        Redirect / https://md.freifunk.net/
</VirtualHost>
```

Changelog

- bind-ffmd als Ipv6-Docker-Container aufgesetzt, siehe #13
- Update auf Debian 10.4 — *Stefan Haun 2020-05-10 13:15*
- md.freifunk.net wird jetzt vom Wordpress-Container auf diesem Host ausgeliefert — *Stefan Haun 2020-06-12 22:01*
- Updaze auf Debian 10.8 —*Kwasir 2021-02-10 21:00*

From:
https://wiki.netz39.de/ - **Netz39**

Permanent link:
https://wiki.netz39.de/freifunk:server:web

Last update: **2021-02-11 20:00**