Docker network für kvwmap

Dr.-Ing. Peter Korduan, GDI-Service Rostock

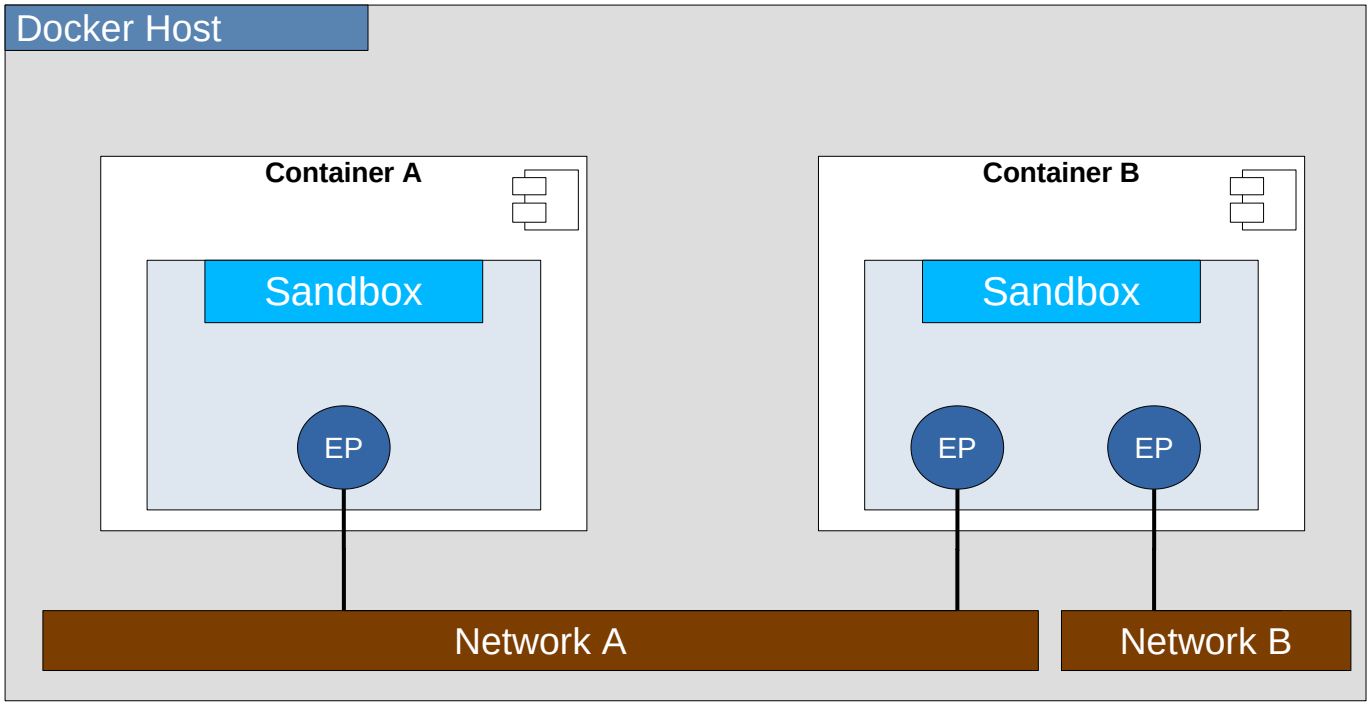
Greifswald, 29.09.2021

Wozu Docker network für kvwmap

- Problemstellung
  - Alle Anwendungen liefen unter einem Apache (kvwmap, kvwmap_dev, 3rdparties, geoserver, owncloud, phpMyAdmin, ...
  - Konfiguration der Container war nicht standardisiert Eintragungen in env_and_volumes
  - Test-Container mußten anderen Namen haben als die der Produktion
  - Anpassungen in den Konfigurationen beim Wechsel zwischen Test, Develop und Produktion
  - Adressbereich von Containern war nicht festgelegt
  - Netzwerkcode war früher im docker-Deamon jetzt in externer Bibliothek

Was ist Docker network

- Docker Netzwerk basiert auf:
  - Container Network Model (CNM): Architekturmodell
  - libnetwork: Implementierung der Modells
  - Treiber: Zur Umsetzung bestimmter Topologien



Sandbox
isolierter Netzwerkstapel

Endpoints
virtuelle Ethernet Schnittstelle

Network
virtuelle Switch (bridge)

kvwmap nutzt Single-host bridge networks

- Netzwerke existieren nur im Docker-Host
- Container können nur auf andere Container innerhalb des Host zugreifen
- Neue Container sind automatisch mit default Network bridge verbunden
- $ docker network ls
  - Zeigt existierende Netzwerke

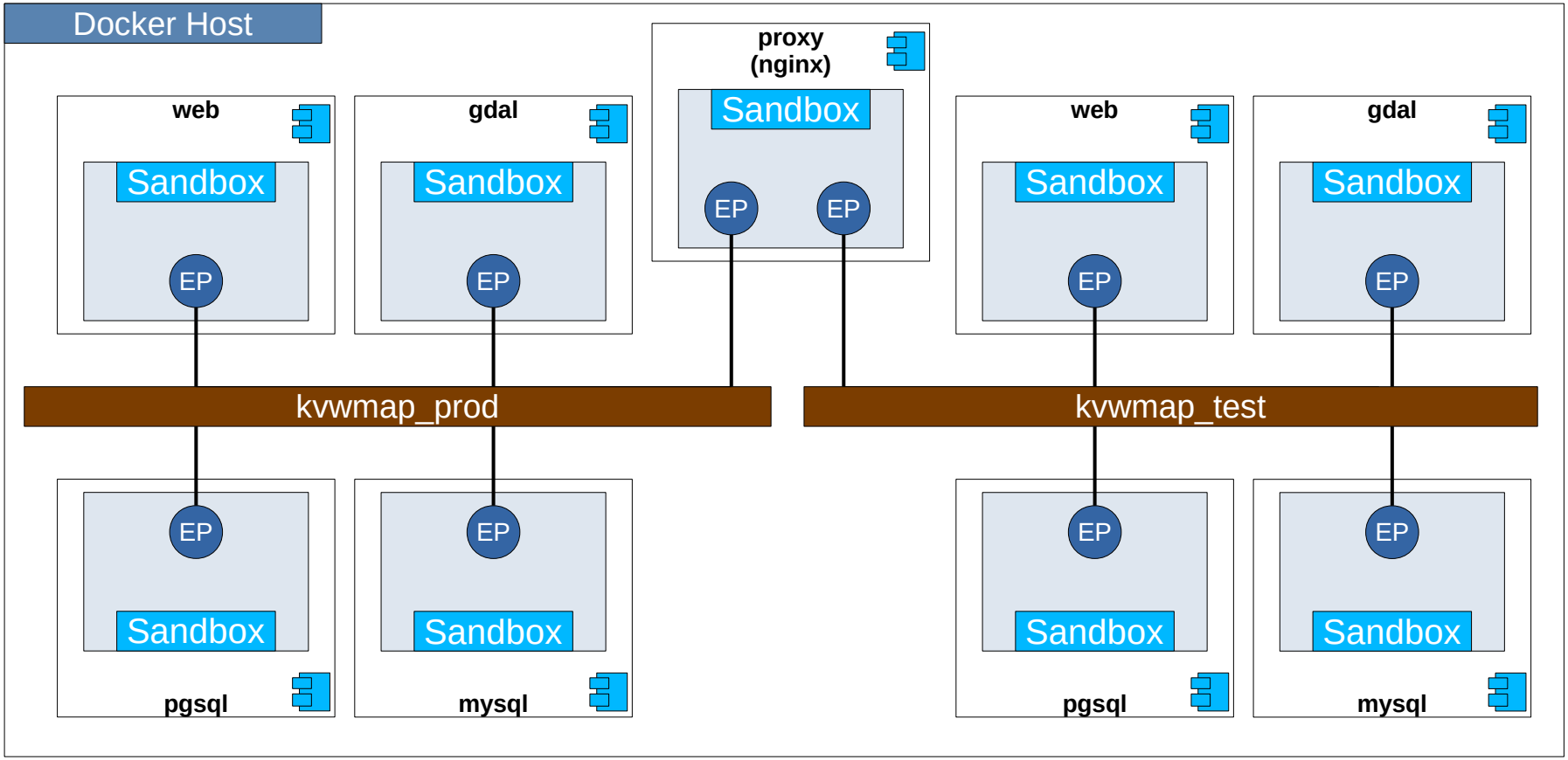
```
gisadmin@h2754512:~ $ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
dff58c47305f        bridge              bridge              local
```

Je ein Netzwerk für jede Umgebung

- $ docker network create -d bridge kvwmap_prod
- $ docker network create -d bridge kvwmap_dev
- $ docker network create -d bridge kvwmap_demo

```
gisadmin@h2754512:~ $ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
dff58c47305f        bridge              bridge              local
ee2d34bd099f        host                host                local
24005a95490d        kvwmap_demo         bridge              local
f70d10803a8a        kvwmap_dev          bridge              local
5f090c5fedc1        kvwmap_prod         bridge              local
6e938214d00b        none                null                local
```

kvwmap Netzwerke



docker compose

- Neue Dateipfad Konvention
  - Alles unter /home/gisadmin/networks
  - pro Network ein Unterverzeichnis

```
[gisadmin@h2754512:~ $ cd networks
[gisadmin@h2754512:~/networks $ ll
insgesamt 8
drwxrwxr-x 6 gisadmin gisadmin 4096 Jul 29 14:55 kvwmap_demo
drwxrwxr-x 6 gisadmin gisadmin 4096 Jul 29 13:32 kvwmap_prod
[gisadmin@h2754512:~/networks $ cd kvwmap_prod
[gisadmin@h2754512:~/networks/kvwmap_prod $ ll
insgesamt 20
-rw-rw-r-- 1 gisadmin gisadmin 3824 Jul 29 13:32 docker-compose.yaml
drwxrwxr-x 5 gisadmin gisadmin 4096 Jul  9 18:12 mysql
drwxrwxr-x 4 gisadmin gisadmin 4096 Jul 23 11:50 pgsql
drwxrwxr-x 2 gisadmin gisadmin 4096 Jul  9 16:23 proj
drwxrwxr-x 8 gisadmin gisadmin 4096 Jul 11 19:59 web
```

- Container Konfiguration pro Netzwerk in einer Datei

  /home/gisadmin/networks/
  kvwmap_prod/docker-compose.yaml

docker-compose.yaml

- Definition Netzwerk und IP-Bereich
- Container als Services
- name
- image
- network
- alias
- env-ironment
- volumes

```yaml
version: "3.6"
networks:
  kvwmap_prod:
    name: kvwmap_prod
    ipam:
      driver: default
      config:
        - subnet: "172.0.10.0/24"
```

```yaml
version: "3.6"
networks:
  kvwmap_prod:
    name: kvwmap_prod
    ipam:
      driver: default
      config:
        - subnet: "172.0.10.0/24"
services:
  web:
    hostname: kvwmap_prod-web
    image: pkorduan/kvwmap-server:2.2.3
    networks:
      kvwmap_prod:
        aliases:
          - web
    environment:
      # Environment variables with only a key are resolved to their values on the machine Compose is running on, which
      - OS_USER=gisadmin
      - IP_EXTERN=$IP_EXTERN
      - DOMAIN_EXTERN=$DOMAIN_EXTERN
      - KVWMAP_INIT_PASSWORD=$KVWMAP_INIT_PASSWORD
      - TERM=linux
      - COLUMNS=180
      - LINES=200
    volumes:
      - "/home/gisadmin/networks/kvwmap_prod/pgsql/.pgpass:/root/.pgpass"
      - "/home/gisadmin/networks/kvwmap_prod/pgsql/.pgpass_gisadmin:/home/gisadmin/.pgpass"
      - "/home/gisadmin/networks/kvwmap_prod/web/apache2/sites-available:/etc/apache2/sites-available"
      - "/home/gisadmin/networks/kvwmap_prod/web/apache2/sites-enabled:/etc/apache2/sites-enabled"
      - "/home/gisadmin/networks/kvwmap_prod/web/cron/load_gisadmin_cron_file:/etc/cron.hourl/load_gisadmin_cron_file"
      - "/home/gisadmin/networks/kvwmap_prod/web/php/7.3:/etc/php/7.3"
      - "/home/gisadmin/networks/kvwmap_prod/web/phpmyadmin/config.inc.php:/srv/www/phpmyadmin/config.inc.php"
      - "/home/gisadmin/networks/kvwmap_prod/proj/epsg:/usr/share/proj/epsg"
      - "/home/gisadmin/networks/kvwmap_prod/proj/MVTR2010.gsb:/usr/share/proj/MVTR2010.gsb"
      - "/home/gisadmin/networks/kvwmap_prod/proj/MVTRS4283.gsb:/usr/share/proj/MVTRS4283.gsb"
      - "/home/gisadmin/networks/kvwmap_prod/web/www:/var/www"
```

Proxy Container

- unter /home/gisadmin/proxy
- Konfiguration in docker-compose.yaml

```
gisadmin@gdi-service:~/proxy $ ll
total 24
-rw-r--r-- 1 gisadmin gisadmin  493 Aug 12 08:34 docker-compose.yaml
-rwxr-xr-x 1 gisadmin gisadmin 2723 Aug 12 08:34 init-letsencrypt.sh
drwxr-xr-x 3 root     root     4096 Aug 12 08:38 letsencrypt
drwxr-xr-x 2 root     root     4096 Aug 12 08:38 logs
drwxr-xr-x 3 gisadmin gisadmin 4096 Aug 12 08:34 nginx
drwxr-xr-x 3 gisadmin gisadmin 4096 Aug 12 08:36 www
```

- container nginx und certbot
- Mapping nach 80 und 443

```
version: '3'
services:
  nginx:
    image: nginx:stable
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./nginx:/etc/nginx
      - ./letsencrypt:/etc/letsencrypt
      - ./www:/var/www/html
      - ./logs:/var/log/nginx

  certbot:
    image: certbot/certbot
    entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew; sleep 24h & wait $${!}; done;'"
    volumes:
      - ./letsencrypt:/etc/letsencrypt
      - ./www:/var/www
      - ./logs:/var/logs/letsencrypt
```

- docker network connect kvwmap_prod proxy

dcm für network

- dcm <cmd> <container> „ohne“ ohne Netzwerk
- dcm <cmd> <container> [network] mit Netzwerk
- Wenn kein network angegeben wird: kvwmap_prod
  - dcm run web ohne
  - dcm rerun web
  - dcm rerun web kvwmap_dev
- Container werden beim Starten automatisch dem entsprechenden Netzwerk zugeordnet
- Wenn es nicht existiert wird es vorher angegelegt
  - dcm run proxy Startet proxy und connected zu allen laufenden Netzwerken

Port mapping

- $ docker container run -d –name proxy \
  --network kvwmap_prod \
  --publish 80:80 \
  nginx
- $ docker port web

```
gisadmin@h2754512:~ $ docker port proxy
443/tcp -> 0.0.0.0:443
80/tcp -> 0.0.0.0:80
```

- Nur nginx Container ist nach außen connected über Port 80
- nginx fungiert als proxy zu containern in den verschiedenen Netzwerken
- pgsql mapping

```
gisadmin@h2754512:~ $ docker port kvwmap_prod_pgsql_1
5432/tcp -> 0.0.0.0:5432
```

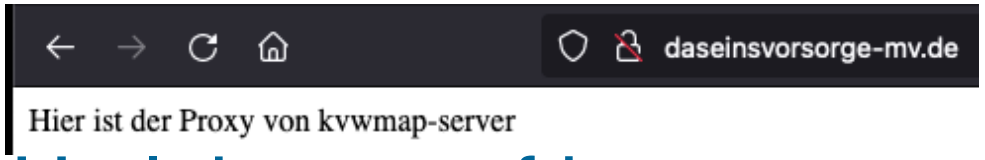
Neue dcm Befehle für proxy

- $ dcm run proxy
- $ dcm stop proxy
- $ dcm rm proxy
- $ dcm rerun proxy
- $ dcm reload proxy (Reload nginx config)
- $ dcm console proxy (Wechselt in nginx-Container)

```
run_proxy_container() {
  echo "Starte proxy Container"
  docker-compose -f $USER_DIR/proxy/docker-compose.yaml up -d nginx
}
```

Proxy Konfiguration – https Umleitung

- /home/gisadmin/proxy/nginx/default.conf
- http://daseinsvorsorge-mv.de



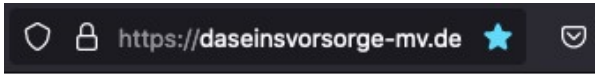
- Umleitung auf https

```
location / {
    return 301 https://$host$request_uri;
}
```

```
server {
    listen       80;
    listen  [::]:80;
    server_name  daseinsvorsorge-mv.de;

    root /var/www/html;

    location / {
        #return 301 https://$host$request_uri;
    }

    location /.well-known/ {

    }

    error_page  500 502 503 504  /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }
}
```

- dcm reload proxy
- Umleitung

  von: http://daseinsvorsorge-mv.de
  nach: https://daseinsvorsorge-mv.de



- Nutzt: ./proxy/nginx/default-ssl.conf

Proxy Konfiguration – Startseite

- /home/gisadmin/proxy/nginx/default-ssl.conf
- Wurzel “/“ verweist auf web-Container im Netzwerk kvwmap_prod
- proxy_pass http://kvwmap_prod_web_1

```
server {
  listen 443 ssl;
  server_name daseinsvorsorge-mv.de;

  root   /var/www/html;

  ssl_certificate /etc/letsencrypt/live/daseinsvorsorge-mv.de/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/daseinsvorsorge-mv.de/privkey.pem;

  include /etc/letsencrypt/options-ssl-nginx.conf;
  ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

  location /.well-known/ {
  }

  location / {
    proxy_pass http://kvwmap_prod_web_1/;
  }
```

Proxy-Konfiguration – Web apps

- Weitere config-Dateien inkludiert in /home/gisadmin/proxy/nginx/default-ssl.conf

```
    include /etc/nginx/sites-enabled/*.conf;
}
```

- alle conf-Dateien in sites-enabled wie bei Apache

```
gisadmin@daseinsvorsorge-mv:~/proxy/nginx $ ls -l sites-available/
total 8
-rw-r--r-- 1 gisadmin gisadmin  94 Sep 28 14:35 kvwmap_test.conf
-rw-r--r-- 1 gisadmin gisadmin 241 Aug 19 12:13 README.md
gisadmin@daseinsvorsorge-mv:~/proxy/nginx $ ls -l sites-enabled/
total 4
lrwxrwxrwx 1 gisadmin gisadmin  35 Sep 28 14:35 kvwmap_test.conf -> ../sites-available/kvwmap_test.conf
-rw-r--r-- 1 gisadmin gisadmin 169 Aug 19 12:13 README.md
```

- proxy_pass http://kvwmap_prod_test_1/
- dcm reload proxy

```
location /test/ {
    proxy_pass http://kvwmap_test_web_1/;
}
```

Nächster Schritt: kvwmap_test_web_1 anlegen

Neue Umgebung Test anlegen

- $ dcm install network test
- Kopiert kvwmap-server/networks/kvwmap_prod nach networks/kvwmap_test
- Anpassungen in kvwmap_test/docker-compose.yaml
  - Ersetzt kvwmap_prod => kvwmap_test
  - IP-Range für Netzwerk + 10 (172.0.10.0 => 172.0.20.0)
  - Port nach außen für pgsql + 1 (5432 => 5433)
- Inhalte von kvwmap_prod übernehmen (oder andere)

Inhalte von /home/gisadmin übernehmen

- Kopieren der Ressourcen des web-Containers
  cp -Rp $USER_DIR/www/apps/* $USER_DIR/networks/$network_name/web/www/apps
  cp -Rp $USER_DIR/www/cron/* $USER_DIR/networks/$network_name/web/www/cron
  cp -Rp $USER_DIR/www/data/* $USER_DIR/networks/$network_name/web/www/data
- Einstellen der Rechte des mysql-Nutzers kvwmap auf die neue Netzwerkmaske
  - host 172.17.% => 172.0.10.% (ggf. Subnetz anpassen 20, 30 etc.)
- Kopieren der Ressourcen des mysql-Containers
  cp -Rp $USER_DIR/db/mysql/* $USER_DIR/networks/$network_name/mysql/data
  cp -Rp $USER_DIR/etc/mysql/* $USER_DIR/networks/$network_name/mysql/etc
  cp -Rp $USER_DIR/www/logs/mysql/* $USER_DIR/networks/$network_name/mysql/logs
- Kopieren der Ressourcen des pgsql-Containers
  cp -Rp $USER_DIR/db/postgresql/data/* $USER_DIR/networks/$network_name/pgsql/data
  cp -Rp $USER_DIR/etc/postgresql/.pgpass $USER_DIR/networks/$network_name/pgsql
  cp -Rp $USER_DIR/www/logs/pgsql/* $USER_DIR/networks/$network_name/pgsql/logs

Container im Netzwerk starten

- Proxy zunächst ohne https konfigurieren
  - in proxy/nginx/default.conf

    return 301 https://$host$request_uri; auskommentieren
  - in proxy/nginx/nginx.conf
  - include /etc/nginx/default-ssl.conf; auskommentieren
- docker-compose run certbot ausführen
  - siehe Kommentar im docker-compose.yaml
- $ dcm run all test
  - Passwörter für mysql root und postgres Nutzer festlegen falls es neue leere Datenbanken sind
- Proxy auf https einstellen

Container in mehreren Netzwerken

- $ docker ps -a

```
PORTS                                          NAMES
0.0.0.0:5433->5432/tcp                         kvwmap_test_pgsql_1
8080/tcp                                       kvwmap_test_gdal_1
3306/tcp                                       kvwmap_test_mysql_1
80/tcp, 443/tcp                                kvwmap_test_web_1
80/tcp, 443/tcp                                kvwmap_prod_web_1
8080/tcp                                       kvwmap_prod_gdal_1
0.0.0.0:5432->5432/tcp                         kvwmap_prod_pgsql_1
3306/tcp                                       kvwmap_prod_mysql_1
0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp       proxy_nginx_1
```

- $ docker network ls

```
NETWORK ID     NAME            DRIVER    SCOPE
b5baf9d525de   bridge          bridge    local
ae7dc7607121   host            host      local
a184e3f64787   kvwmap_prod     bridge    local
c24429cf9cea   kvwmap_test     bridge    local
c626531f3d4c   none            null      local
c2c4a11c5493   proxy_default   bridge    local
```

- $ docker network connect kvwmap_test proxy_nginx_1
- $ docker inspect -f {{.NetworkSettings.Networks}} proxy_nginx_1

```
map[kvwmap_prod:0xc000034000 kvwmap_test:0xc0000346c0 proxy_default:0xc000034d80]
```

kvwmap in anderen Netzwerken

- Quellcode:

  /home/gisadmin/networks/kvwmap_test/web/www/apps/kvwmap

- Daten

  /home/gisadmin/networks/kvwmap_test/web/www/data

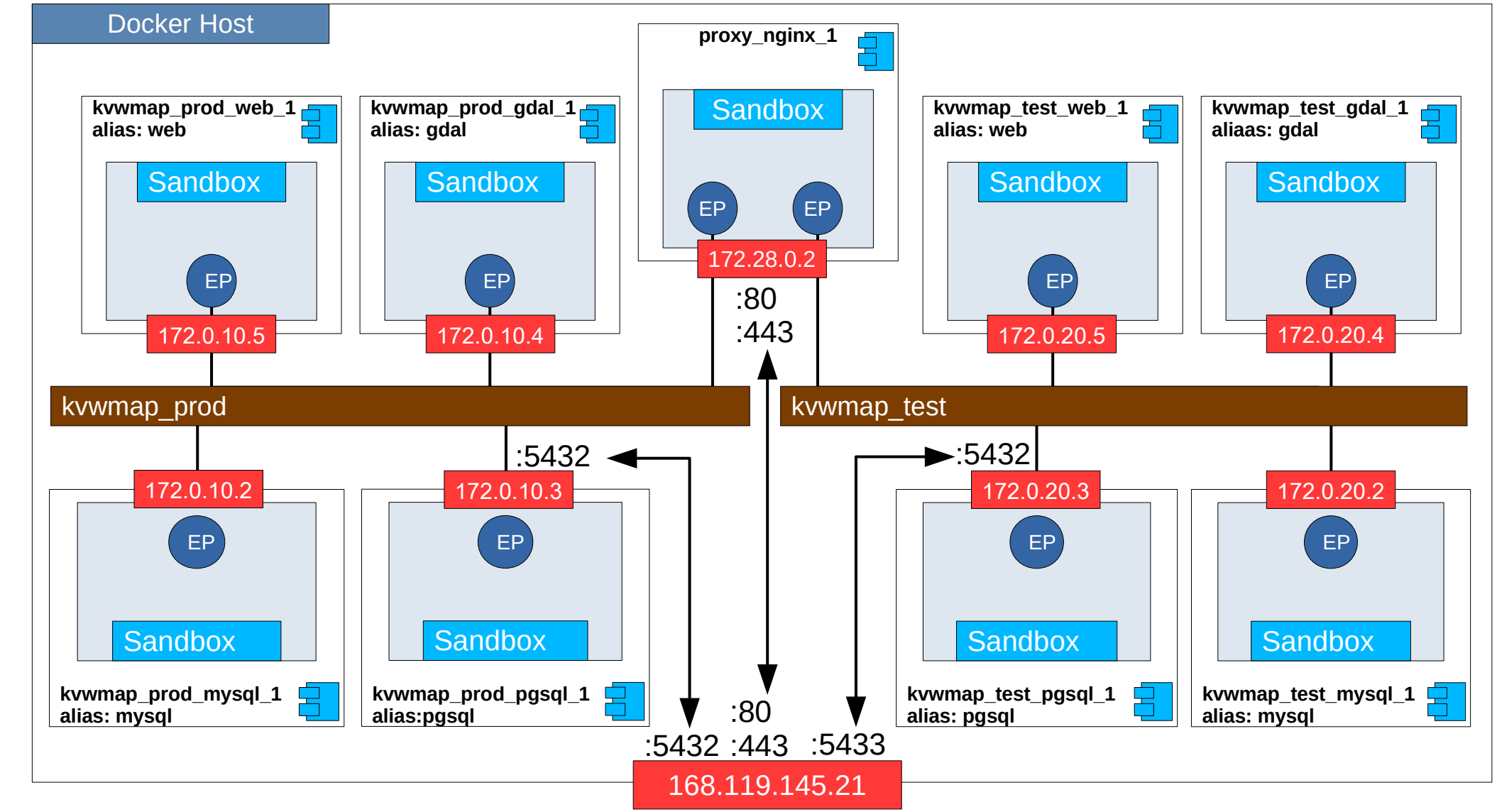
- Logs

  /home/gisadmin/networks/kvwmap_test/web/www/logs

- URL:

  https://daseinsvorsorge-mv.de/test/

Aliase in anderen Netzwerken



Sichtbarkeit im Netzwerk

```
gisadmin@daseinsvorsorge-mv:~/networks/kvwmap_test/web/www $ dcm console web
Load config file /home/gisadmin/kvwmap-server/config/config
Folgenden Hostname ermittelt: daseinsvorsorge-mv.de
Folgenden Domainname ermittelt: daseinsvorsorge-mv.de
Folgende IP ermittelt: 168.119.145.21
Öffne ein Terminal zum web container
docker-compose -f /home/gisadmin/networks/kvwmap_prod/docker-compose.yaml exec web /bin/bash
root@web-prod:/home/gisadmin $ ping mysql
PING mysql (172.0.10.2) 56(84) bytes of data.
64 bytes from kvwmap_prod_mysql_1.kvwmap_prod (172.0.10.2): icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from kvwmap_prod_mysql_1.kvwmap_prod (172.0.10.2): icmp_seq=2 ttl=64 time=0.121 ms
^C
--- mysql ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 0.048/0.084/0.121/0.037 ms
root@web-prod:/home/gisadmin $ exit
exit
gisadmin@daseinsvorsorge-mv:~/networks/kvwmap_test/web/www $ dcm console web test
Load config file /home/gisadmin/kvwmap-server/config/config
Folgenden Hostname ermittelt: daseinsvorsorge-mv.de
Folgenden Domainname ermittelt: daseinsvorsorge-mv.de
Folgende IP ermittelt: 168.119.145.21
Öffne ein Terminal zum web container
docker-compose -f /home/gisadmin/networks/kvwmap_test/docker-compose.yaml exec web /bin/bash
root@web_kvwmap_test:/home/gisadmin $ ping mysql
PING mysql (172.0.20.2) 56(84) bytes of data.
64 bytes from kvwmap_test_mysql_1.kvwmap_test (172.0.20.2): icmp_seq=1 ttl=64 time=0.079 ms
64 bytes from kvwmap_test_mysql_1.kvwmap_test (172.0.20.2): icmp_seq=2 ttl=64 time=0.120 ms
^C
--- mysql ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 0.079/0.099/0.120/0.022 ms
```