Dockerized PHP

2018 Stütz

PHPDocker.io - PHP and Docke

https://phpdocker.io

PHPDocker.io beta

News Generator Contact

What is PHPDocker.io?

PHPDocker.io is a tool that will help you build a typical **PHP development environment** based on **Docker** with just a few clicks. It supports provisioning of the usual services (MySQL/MariaDB, Redis, Elasticsearch...), with more to come. PHP 7.1 is supported, as well as 7.0 and 5.6.

Why Docker? Containers are typically faster to set up, less prone to provisioning errors (pre-setup images), and last but not least, web app deployment is going the way of the container. **PHP applications** work really well on **Docker**, and, as applications are increasingly set up for Docker, so should our development environments.

News

Interview with Karl Hughes from Shiphp

November 15th, 2017

Karl Hughes from Shiphp **interviewed me for his blog** just this week.

Continue reading >

(1) comment(s)

June updates: Clickhouse

June 29th, 2017

June has been pretty quiet. Mostly, under-the-hood improvements like optimisations to the generated docker-compose file for larger projects (thanks to **z1r0-**).

A new, interesting feature is **Clickhouse** support, kindly contributed by **Andrey Bolonin**. I'm not familiar with Clickhouse, a SQL-esque data store for larger, if not huge, datasets for analysis by Yandex, but it's intriguing.

Continue reading >

(0) comment(s)

PHPDocker.io - Generator

https://phpdocker.io/generator

PHPDocker.io beta

News Generator Contact

Global configuration

Project name: Used on host, container, vm and folder names

Base port: For nginx, Mailhog control panel...

Application type: Generic: Symfony 4, Zend, Laravel, Lumen...

- ✓ Generic: Symfony 4, Zend, Laravel, Lumen...
- Symfony 2/3
- Phalcon 3
- Silex

Max upload ...MB): 100

PHP configu...

PHP Version: 7.2.x

Extensions (PHP 7.2.x): None selected

Search

- Memcached
- MySQL
- PostgreSQL
- Redis
- SQLite3
- XDebug

Off Add git (eg for composer)

Please note:

- *The following extensions are already included on the base image: APC, cURL, JSON, MCrypt (sodium in PHP>=7.1), MBString, OPCache, Readline, XML and Zip.*
- *Each PHP version supports a different set of extensions to the others.*
- *Adding git to the container adds ~75MB to it.*

MySQL

klingt vielversprechend, derzeit aber noch nicht einsatzbereit (Silex abgekündigt, Symfony4 fehlt)

Quellen

- http://geekyplatypus.com/dockerise-your-php-application-with-nginx-and-php7-fpm/

- https://sebastianbrosch.blog/2018/docker-stack-nginx-mit-php-7-2-und-mysql/

- https://semaphoreci.com/community/tutorials/dockerizing-a-php-application

Iteration 1

docker-compose.yml

```yaml
web:
    image: nginx:latest
    ports:
    - "8080:80"
```

```
Toms-MBP:my_php_docker stuetz$ docker-compose up -d
Pulling web (nginx:latest)...
latest: Pulling from library/nginx
f2aa67a397c4: Already exists
3c091c23e29d: Pull complete
4a99993b8636: Pull complete
Digest: sha256:0fb320e2a1b1620b4905facb3447e3d84ad36da0b2c8aa8fe3a5a81d1187b884
Status: Downloaded newer image for nginx:latest
Creating my_php_docker_web_1 ... done
Toms-MBP:my_php_docker stuetz$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                  NAMES
8e453a55015f        nginx:latest        "nginx -g 'daemon of…"   20 seconds ago      Up 20 seconds       0.0.0.0:8080->80/tcp   my_php_docker_web_1
```

localhost:8080

Welcome to nginx!

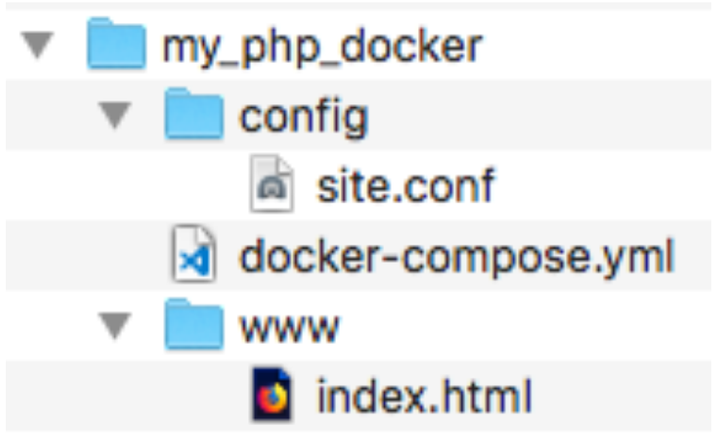
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

```
Toms-MBP:my_php_docker stuetz$ docker-compose down
Stopping my_php_docker_web_1 ... done
Removing my_php_docker_web_1 ... done
```

Iteration 2



docker-compose.yml

```yaml
version: "3"

services:
    web:
        image: nginx:1.14.0-alpine
        ports:
            - "80:80"
        volumes:
            - ./www:/www
            - ./config/site.conf:/etc/nginx/conf.d/default.conf
```

Hier wird alpine als leichtgewichtiges OS verwendet und der Port 80

site.conf

```
server {
    index index.html;
    root /www;
}
```

Hier wird www als root-Verzeichnis für die html-Dateien eingetragen und index.html als index-Datei

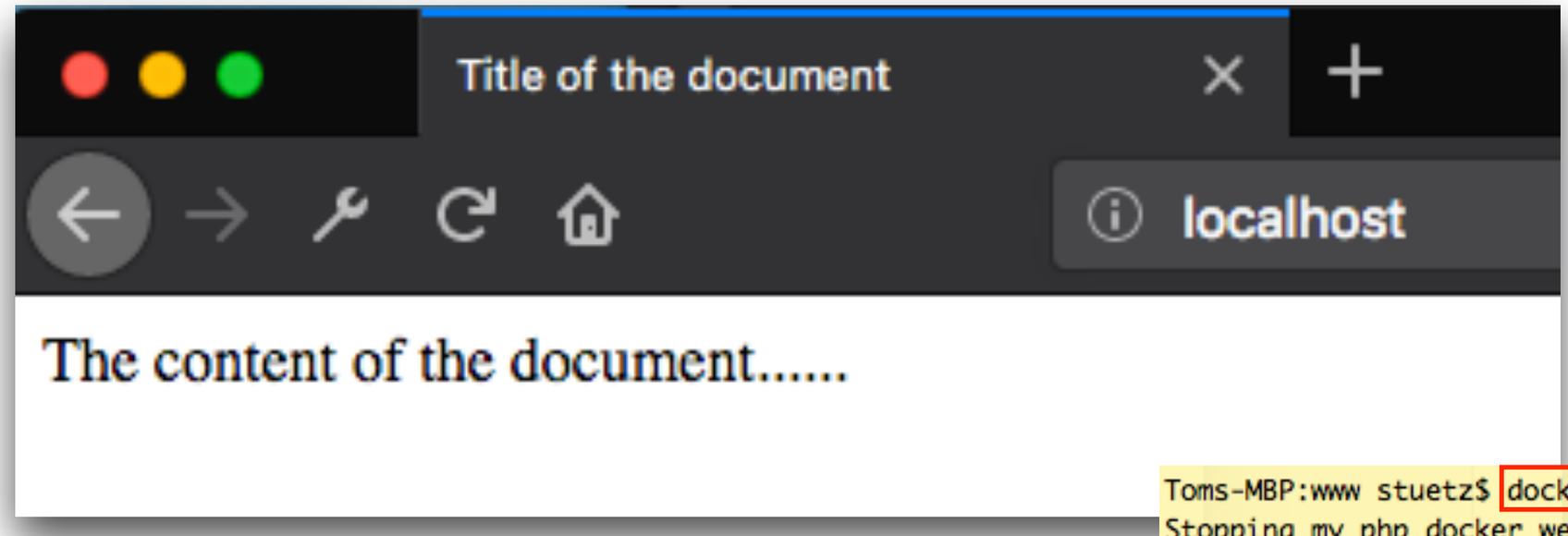
index.html

```html
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document......
</body>

</html>
```

Iteration 2

```
Toms-MBP:my_php_docker stuetz$ docker-compose up -d
Creating network "my_php_docker_default" with the default driver
Pulling web (nginx:1.14.0-alpine)...
1.14.0-alpine: Pulling from library/nginx
ff3a5c916c92: Already exists
ff6bd9dc511a: Pull complete
19deea388932: Pull complete
a778cccae7b5: Pull complete
Digest: sha256:a4bbaf406318769c28ff13fcc464bdf7d43b5ff666e9b5279ce568100499f375
Status: Downloaded newer image for nginx:1.14.0-alpine
Creating my_php_docker_web_1 ... done
```

```
Toms-MBP:www stuetz$ docker ps
CONTAINER ID        IMAGE                 COMMAND                  CREATED             STATUS              PORTS                  NAMES
8235b8796bd8        nginx:1.14.0-alpine   "nginx -g 'daemon of…"   7 minutes ago       Up 2 seconds        0.0.0.0:80->80/tcp     my_php_docker_web_1
```




```
Toms-MBP:www stuetz$ docker-compose down
Stopping my_php_docker_web_1 ... done
Removing my_php_docker_web_1 ... done
Removing network my_php_docker_default
```

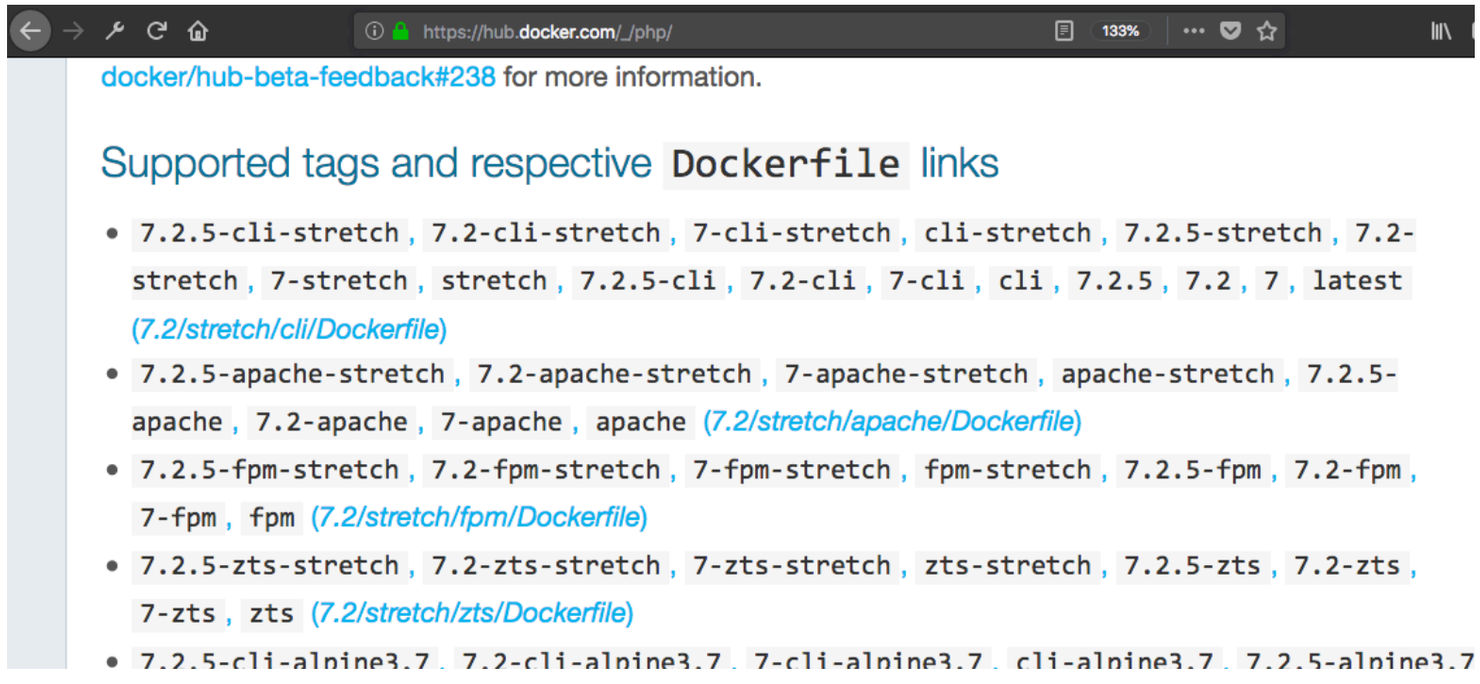
Iteration 3: Einrichten von PHP

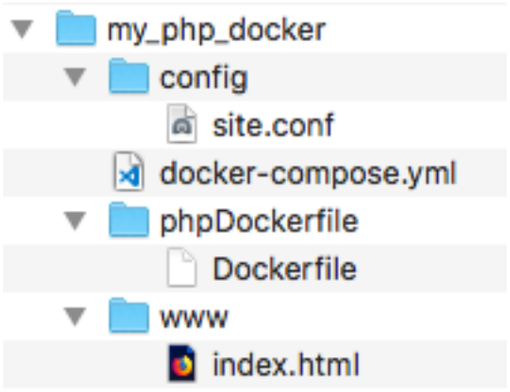
Dockerfile ●

```
1   FROM php:7.2.5-fpm
2
3   RUN docker-php-ext-install -j$(nproc) mysqli
4   RUN docker-php-ext-install -j$(nproc) pdo
5   RUN docker-php-ext-install -j$(nproc) pdo_mysql
```

ein Dockerfile zum Einrichten von php wird erstellt

Im offiziellen php Dockerhub Repo kann man sich alle Versionen ansehen



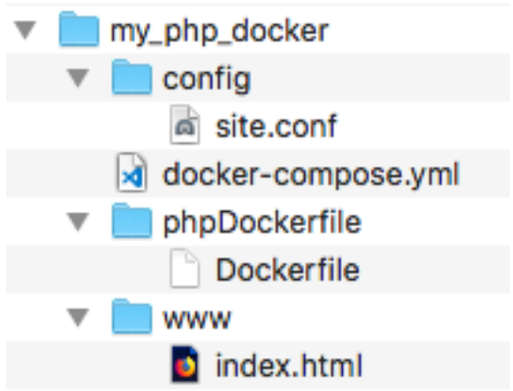


Iteration 3

docker-compose.yml

```yaml
version: "3"

services:
    web:
        image: nginx:1.14.0-alpine
        ports:
            - "80:80"
        volumes:
            - ./www:/www
            - ./config/site.conf:/etc/nginx/conf.d/default.conf
    php:
        build:
            context: ./phpDockerfile/
        ports:
            - 9000:9000
        volumes:
            - ./www:/www
```

„context“ gibt den Pfad zum Dockerfile an. 9000 ist der Port von fastCGI



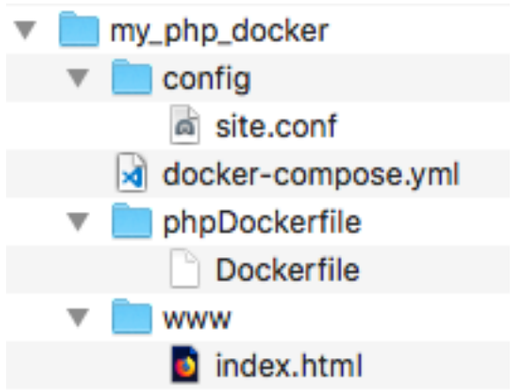
Iteration 3

site.conf

```
1   server {
2       index index.html index.php;
3       root /www;
4
5       location ~ \.php$ {
6           try_files $uri =404;
7           fastcgi_split_path_info ^(.+\.php)(/.+)$;
8           fastcgi_pass php:9000;
9           fastcgi_index index.php;
10          include fastcgi_params;
11          fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
12          fastcgi_param PATH_INFO $fastcgi_path_info;
13      }
14  }
```

Name des php-Containers

Das Config-File wird noch ergänzt, damit der WebServer auf php zugreifen kann



Iteration 3

docker-compose starten

```
Toms-MBP:my_php_docker stuetz$ docker-compose up -d
```

laufende Container

```
Toms-MBP:my_php_docker stuetz$ docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS                    NAMES
4952d3fa50d9        my_php_docker_php      "docker-php-entrypoi…"   About a minute ago  Up About a minute   0.0.0.0:9000->9000/tcp   my_php_docker_php_1
8a590640853f        nginx:1.14.0-alpine    "nginx -g 'daemon of…"   About a minute ago  Up About a minute   0.0.0.0:80->80/tcp       my_php_docker_web_1
```

Images

```
Toms-MBP:my_php_docker stuetz$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
my_php_docker_php   latest              22b3e63048dd        2 minutes ago       367MB
mariadb             10.3                802f90d5da0d        2 weeks ago         401MB
nginx               1.14.0-alpine       d86442b1a2d5        2 weeks ago         18MB
```

durch unser Dockerfile wurde dieses Image erstellt

docker-compose beenden

```
Toms-MBP:my_php_docker stuetz$ docker-compose down --rmi local
Stopping my_php_docker_php_1 ... done
Stopping my_php_docker_web_1 ... done
Removing my_php_docker_php_1 ... done
Removing my_php_docker_web_1 ... done
Removing network my_php_docker_default
Removing image my_php_docker_php
```

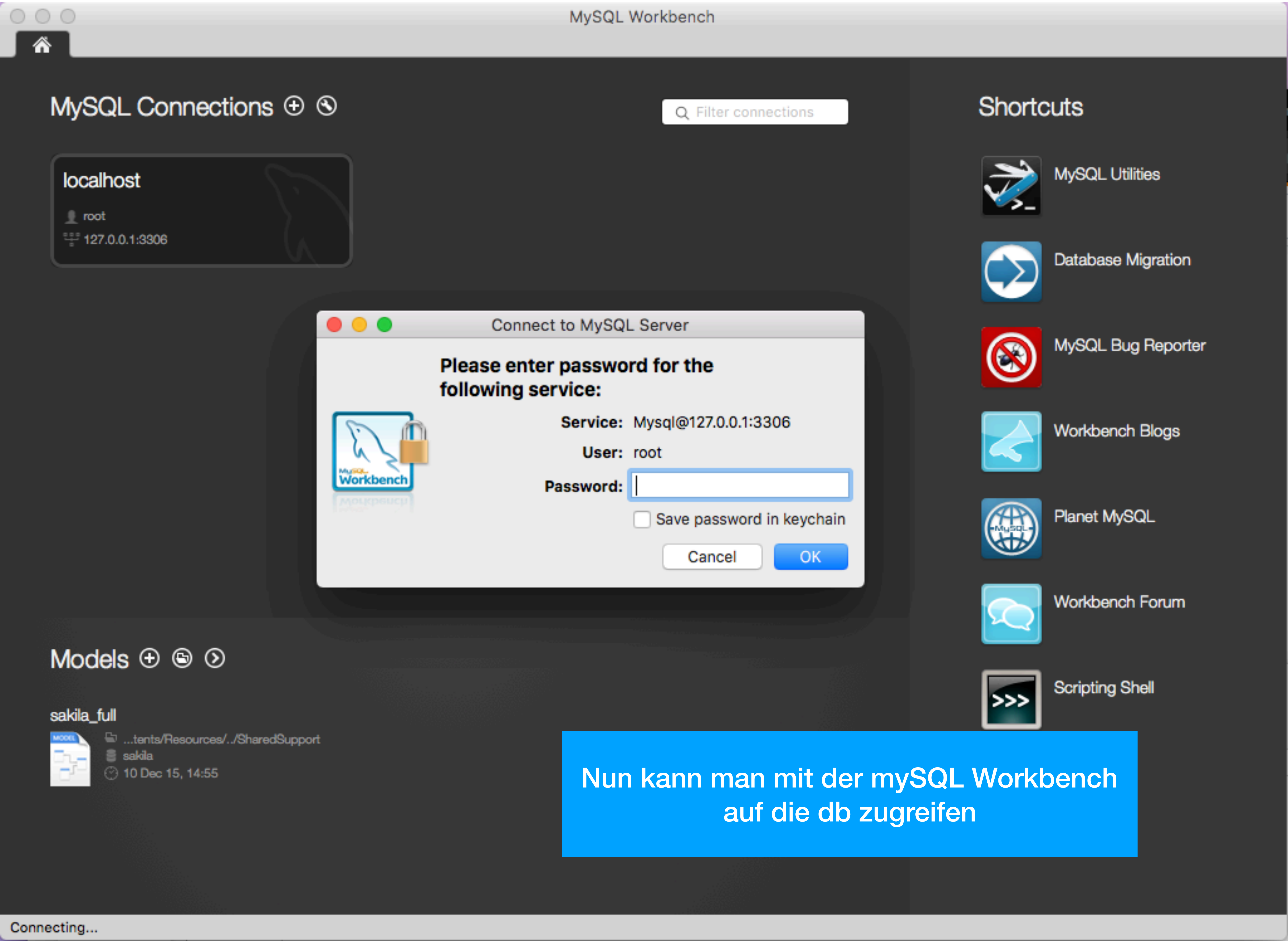
--rmi local löscht das erstellte Image

Iteration 4: Datenbank

docker-compose.yml

```yaml
version: "3"

services:
    web:
        image: nginx:1.14.0-alpine
        ports:
            - "80:80"
        volumes:
            - ./www:/www
            - ./config/site.conf:/etc/nginx/conf.d/default.conf
    php:
        build:
            context: ./phpDockerfile/
        ports:
            - 9000:9000
        volumes:
            - ./www:/www
    mariadb:
        image: mariadb:10.3
        ports:
            - 3306:3306
        environment:
            MYSQL_ROOT_PASSWORD: passme
```

Wir verwenden anstelle von mysql die kompatible mariadb



Nun kann man mit der mySQL Workbench auf die db zugreifen

MySQL Workbench

localhost - Warning - not supported

MANAGEMENT
- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE
- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE
- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

- mydb
  - Tables
  - Views
  - Stored Procedures

Object Info | Session

Schema: mydb

Query 1 | Administration - Server Status

Connection Name
localhost

Host:	6960a825d
Socket:	/var/run/my
Port:	3306
Version:	10.3.6-Maria mariadb.org
Compiled For:	debian-linux
Configuration File:	/etc/my.cnf
Running Since:	Thu May 24

Refresh

Available Server Features

Performance Schema:	Off	PAM Authentication:	
Thread Pool:	n/a	Password Validation:	
Memcached Plugin:	n/a	Audit Log:	
Semisync Replication Plugin:	Off	Firewall:	
SSL Availability:	On	Firewall Trace:	

Server Directories

Base Directory:	/usr
Data Directory:	/var/lib/mysql/
Disk Space in Data Dir:	Could not determine
Plugins Directory:	/usr/lib/mysql/plugin/
Tmp Directory:	/tmp
Error Log:	Off

Server Status
Running

Load
2.13

Connections
4

Traffic
14.93 KB/s

Key Efficiency
66.7%

Selects per Second
0

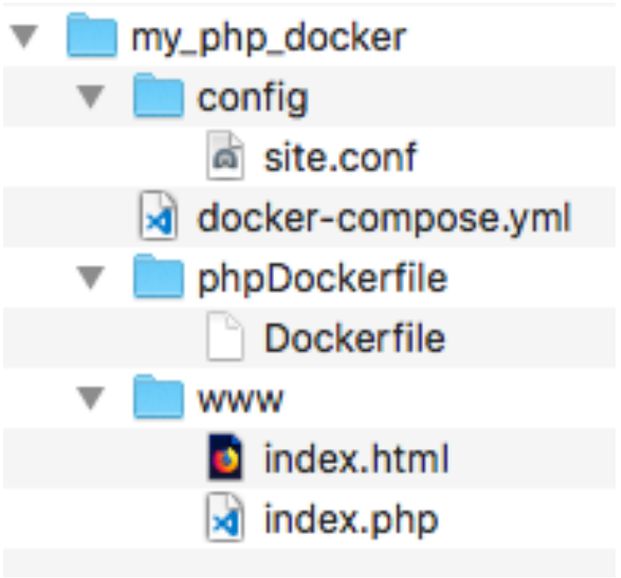
InnoDB Buffer Usage
1.9%

InnoDB Reads per Second
0

InnoDB Writes per Second
0

Management support for target host enabled successfully.

phpinfo



Legt man eine index.php an, mit obigen Inhalt, so kann man den php-Status überprüfen

phpinfo()

localhost/index.php

PHP Version 7.2.5

System	Linux de9da91e6ab2 4.9.87-linuxkit-aufs #1 SMP Wed Mar 14 15:12:16 UTC 2018 x86_64
Build Date	May 5 2018 00:30:20
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--disable-cgi' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--enable-fpm' '--with-fpm-user=www-data' '--with-fpm-group=www-data' 'build_alias=x86_64-linux-gnu'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2

Iteration 5: phpmyadmin

docker-compose.yml

```yaml
version: "3"

services:
    web:
        image: nginx:1.14.0-alpine
        ports:
            - "80:80"
        volumes:
            - ./www:/www
            - ./config/site.conf:/etc/nginx/conf.d/default.conf
    php:
        build:
            context: ./phpDockerfile/
        ports:
            - 9000:9000
        volumes:
            - ./www:/www
    mariadb:
        image: mariadb:10.3
        ports:
            - 3306:3306
        environment:
            MYSQL_ROOT_PASSWORD: passme
    phpmyadmin:
        image: phpmyadmin/phpmyadmin
        links:
            - mariadb:db
        ports:
            - 8000:80
        environment:
            MYSQL_ROOT_PASSWORD: passme
```

Iteration 5

```
Toms-MBP:www stuetz$ docker-compose up -d
Creating network "my_php_docker_default" with the default driver
Creating my_php_docker_php_1     ... done
Creating my_php_docker_mariadb_1 ... done
Creating my_php_docker_web_1     ... done
Creating my_php_docker_phpmyadmin_1 ... done
```

```
Toms-MBP:www stuetz$ docker ps
CONTAINER ID        IMAGE                      COMMAND                  CREATED             STATUS              PORTS                               NAMES
603954409f7f        phpmyadmin/phpmyadmin      "/run.sh phpmyadmin"     5 minutes ago       Up 5 minutes        9000/tcp, 0.0.0.0:8000->80/tcp      my_php_docker_phpmyadmin_1
f27a47e78037        my_php_docker_php          "docker-php-entrypoi…"   5 minutes ago       Up 5 minutes        0.0.0.0:9000->9000/tcp              my_php_docker_php_1
06b893d341aa        nginx:1.14.0-alpine        "nginx -g 'daemon of…"   5 minutes ago       Up 5 minutes        0.0.0.0:80->80/tcp                  my_php_docker_web_1
732a5df4f8e3        mariadb:10.3               "docker-entrypoint.s…"   5 minutes ago       Up 5 minutes        0.0.0.0:3306->3306/tcp              my_php_docker_mariadb_1
```

```
Toms-MBP:www stuetz$ docker images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
my_php_docker_php         latest              44c7639d5313        20 minutes ago      367MB
mariadb                   10.3                802f90d5da0d        2 weeks ago         401MB
nginx                     1.14.0-alpine       d86442b1a2d5        2 weeks ago         18MB
```

```
Toms-MBP:www stuetz$ docker-compose down --rmi local
Stopping my_php_docker_phpmyadmin_1 ... done
Stopping my_php_docker_php_1        ... done
Stopping my_php_docker_web_1        ... done
Stopping my_php_docker_mariadb_1    ... done
Removing my_php_docker_phpmyadmin_1 ... done
Removing my_php_docker_php_1        ... done
Removing my_php_docker_web_1        ... done
Removing my_php_docker_mariadb_1    ... done
Removing network my_php_docker_default
Removing image my_php_docker_php
```

localhost:8000 / db | phpMyAd

localhost:8000/server_databases.php?server=1

phpMyAdmin

Recent Favorites

- New
- information_schema
- mysql
- performance_schema

Server: db

Databases | SQL | Status | User accounts | Export | Import | Settings | More

Databases

Create database

mydb | utf8_general_ci | Create

	Database	Collation	Action
☐	information_schema	utf8_general_ci	Check privileges
☐	mysql	latin1_swedish_ci	Check privileges
☐	performance_schema	utf8_general_ci	Check privileges
	Total: 3	latin1_swedish_ci	

Check all *With selected:* Drop

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

- **Enable statistics**

Console

localhost:8000 / db / mydb / pe

localhost:8000/db_structure.php?server=1&db=mydb&table=person

phpMyAdmin

Recent Favorites

- New
- information_schema
- mydb
  - New
  - person
    - *Columns*
      - New
      - firstname
      - id
      - lastname
- mysql
- performance_schema

Server: db » Database: mydb » Table: person

Browse | Structure | SQL | Search | Insert | Export | Import | More

Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id**	int(11)			No	*None*			Change Drop More
☐	2	**firstname**	varchar(50)	utf8_general_ci		No	*None*			Change Drop More
☐	3	**lastname**	varchar(50)	utf8_general_ci		No	*None*			Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after lastname Go

Indexes

No index defined!

Create an index on 1 columns Go

Partitions

Console

Eigener IP-Bereich

Problem

- Docker benutzt einen eigenen IP-Bereich: 172.17.x.x
- In der Schule wird für das LAN dieser IP-Bereich ebenfalls benutzt
- Daher entsteht ein Konflikt, die Docker-Container sind in der Filge nicht mehr erreichbar

Abhilfe

- Der Docker Host bekommt einen neuen Standard-IP-Bereich.

- Dazu wird die Datei /etc/docker/daemon.json angelegt bzw. geändert:

```
{
  "bip": "172.31.1.1/24",
  "dns": ["10.191.80.6" , "10.191.16.10"]
}
```

Kontrolle

```
stuetz@vm81:~$ sudo ifconfig docker0
docker0   Link encap:Ethernet  HWaddr 02:42:91:93:2a:35
          inet addr:172.31.1.1  Bcast:172.31.1.255  Mask:255.255.255.0
          inet6 addr: fe80::42:91ff:fe93:2a35/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23332 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:30181406 (30.1 MB)  TX bytes:3943275 (3.9 MB)
```

```
stuetz@vm81:~$ sudo docker network inspect bridge
[
    {

        "Name": "bridge",
        "Id": "9df58fc430be8a5d87f8f9087612d2022059209afee8cd98d56849e0442f68f8",
        "Created": "2018-05-31T14:18:19.270297718+02:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.31.1.1/24",
                    "Gateway": "172.31.1.1"
```

docker-compose.yml

```
version: „2"

services:
    web:
        image: nginx:1.14.0-alpine
        ports:
            - "80:80"
        volumes:
            - ./www:/www
            - ./config/site.conf:/etc/nginx/conf.d/default.conf
        networks:
            - myphpnet
    php:
        build:
            context: .
            dockerfile: DockerfilePhp
        ports:
            - 9000:9000
        volumes:
            - ./www:/www
        networks:
            - myphpnet
    mariadb:
        image: mariadb:10.3
        ports:
            - 3306:3306
        environment:
            MYSQL_ROOT_PASSWORD: passme
        networks:
            - myphpnet
    phpmyadmin:
        image: phpmyadmin/phpmyadmin
        links:
            - mariadb:db
        ports:
            - 8000:80
        environment:
            MYSQL_ROOT_PASSWORD: passme
        networks:
            - myphpnet
networks:
    myphpnet:
        driver: bridge
        ipam:
            config:
                - subnet: 172.30.1.1/24
```

Man kann auch noch bei den einzelnen Docker-Maschinen einen eigenen Adressbereich vergeben

Start mit:
docker-compose up -d

Kontrolle

```
stuetz@vm81:~$ sudo docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
9df58fc430be        bridge              bridge              local
a18a1a8202dd        host                host                local
b63728e01fe6        myphpdocker_phpnet  bridge              local
8b8d18e3bc0e        none                null                local
```

Kontrolle im Detail

```
sudo docker network inspect myphpdocker_phpnet
```


```
"Containers": {
    "14a8bfa60382cd7bb6049a2aa03a3ce969bff440ecc06817a6d563a0c43ed8a9": {
        "Name": "myphpdocker_php_1",
        "EndpointID": "522247a8874eb5b51860f86e2bb5c3de9c5196db55ceca05c0e24df5b1fc91ba",
        "MacAddress": "02:42:ac:1e:01:03",
        "IPv4Address": "172.30.1.3/24",
        "IPv6Address": ""
    },
    "3216fca48e23cf7ea6b47df5de34b70b95c2a2fb13331dd00274c767c505ae13": {
        "Name": "myphpdocker_web_1",
        "EndpointID": "1b9351c8146a123634ce30273c2cd893e08244746488ad776c3af9727546be9b",
        "MacAddress": "02:42:ac:1e:01:02",
        "IPv4Address": "172.30.1.2/24",
        "IPv6Address": ""
    },
    "84307e4cb7c6cb84f6dbf6960aa9bc9deb86e8124f3410a23847bbad8c827556": {
        "Name": "myphpdocker_mariadb_1",
        "EndpointID": "9a8472af261bb0af0dc31a016644918486a604002fa8d17bb491361e17070e09",
        "MacAddress": "02:42:ac:1e:01:04",
        "IPv4Address": "172.30.1.4/24",
        "IPv6Address": ""
    },
    "f388d7fefef101a4e4dde4ac0e3231747cedefa705f82e39337351cd4eb40a0a": {
        "Name": "myphpdocker_phpmyadmin_1",
        "EndpointID": "99e63266b891311712fcd13421fd76c21add5598a893ca85547d93f755351647",
        "MacAddress": "02:42:ac:1e:01:05",
        "IPv4Address": "172.30.1.5/24",
        "IPv6Address": ""
    }
},
```