

docker-compose files

hier findet ihr die im Stream verwendeten docker-compose.yml Dateien

- [portainer](#)
- [nginx proxy manager](#)
- [vaultwarden](#)
- [rustdesk](#)
- [guacamole](#)
- [bar assistant](#)

portainer

auf dem Host System muss vor dem ausrollen des Portainer Containers noch ein eigenes Netzwerk sowie ein Volume angelegt werden.

Hierfür müssen folgende Befehle ausgeführt werden:

```
docker network create proxy
```

```
docker volume create portainer
```

ich lege mir im Homeverzeichnis auf meinem Docker Host immer noch einen eigenen Ordner für die docker-compose Datei an

```
mkdir portainer && cd portainer && nano docker-compose.yml
```

hier wird jetzt folgendes eingefügt

```
version: '3.8'

services:
  portainer:
    container_name: portainer
    hostname: portainer
    image: portainer/portainer-ce
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    networks:
      - proxy
    ports:
      - "8000:8000"
      - "9443:9443"
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - portainer:/data
    environment:
```

- TZ=Europe/Berlin

volumes:

portainer:

external: true

networks:

proxy:

external: true

nun kann der Container gestartet werden

```
docker compose up -d
```

nginx proxy manager

version: '3.8'

services:

nginx-proxy-manager:

container_name: nginx-reverse-proxy

hostname: nginx

image: jc21/nginx-proxy-manager:latest

restart: unless-stopped

networks:

- proxy

ports:

- "80:80"

- "81:81"

- "443:443"

volumes:

- ~/nginx_data/data:/data

- ~/nginx_data/letsencrypt:/etc/letsencrypt

environment:

if you want pretty colors in your docker logs:

- FORCE_COLOR=1

- TZ=Europe/Berlin

networks:

proxy:

external: true

vaultwarden

vor dem Ausrollen noch ein Volume mit dem Namen `vaultwarden` anlegen

```
version: '3.8'
```

```
services:
```

```
  vaultwarden:
```

```
    container_name: vaultwarden
```

```
    hostname: vaultwarden
```

```
    image: vaultwarden/server:latest
```

```
    restart: unless-stopped
```

```
    security_opt:
```

```
      - no-new-privileges:true
```

```
    networks:
```

```
      - proxy
```

```
    ports:
```

```
      - "8080:80"
```

```
    volumes:
```

```
      - vaultwarden:/data
```

```
    environment:
```

```
      - TZ=Europe/Berlin
```

```
      - WEBSOCKET_ENABLED=true # Enable WebSocket notifications.
```

```
      - ADMIN_TOKEN=xxxxxxx[]# <-- mit eigenem Token anpassen
```

```
      - DOMAIN=https://xxx.xxx.xx[]# <-- mit eigener Domain anpassen
```

```
      - SIGNUPS_ALLOWED=false
```

```
      - SIGNUPS_VERIFY=true
```

```
volumes:
```

```
  vaultwarden:
```

```
    external: true
```

```
networks:
```

```
  proxy:
```

```
    external: true
```

rustdesk

ein Netzwerk anlegen mit dem Namen `rustdesk`

ein Volume anlegen mit dem Namen `rustdesk`

```
version: '3.8'
```

```
services:
```

```
  hbbs:
```

```
    container_name: rustdesk-hbbs
```

```
    hostname: rustdesk-hbbs
```

```
    image: rustdesk/rustdesk-server:latest
```

```
    restart: unless-stopped
```

```
    networks:
```

```
      - rustdesk
```

```
  ports:
```

```
    - "21115:21115"
```

```
    - "21116:21116"
```

```
    - "21116:21116/udp"
```

```
    - "21118:21118"
```

```
  depends_on:
```

```
    - hbbr
```

```
  command: hbbs -r <EURE.DYNDS.DE>:21117
```

```
  volumes:
```

```
    - rustdesk:/root
```

```
  environment:
```

```
    - TZ=Europe/Berlin
```

```
  hbbr:
```

```
    container_name: rustdesk-hbbr
```

```
    hostname: rustdesk-hbbr
```

```
    image: rustdesk/rustdesk-server:latest
```

```
    restart: unless-stopped
```

```
    networks:
```

```
      - rustdesk
```

```
  ports:
```

```
    - "21117:21117"
```

- "21119:21119"

command: hbbr

volumes:

- rustdesk:/root

environment:

- TZ=Europe/Berlin

networks:

rustdesk:

external: true

volumes:

rustdesk:

external: true

guacamole

vor dem Ausrollen noch ein Volume mit dem Namen `guacamole` anlegen

```
version: '3.8'
```

```
services:
```

```
  guacamole:
```

```
    container_name: guacamole
```

```
    hostname: guacamole
```

```
    image: maxwaldorf/guacamole
```

```
    restart: unless-stopped
```

```
    networks:
```

```
      - proxy
```

```
    ports:
```

```
      - "7070:8080"
```

```
    volumes:
```

```
      - guacamole:/config
```

```
    environment:
```

```
      TZ: Europe/Berlin
```

```
      EXTENSIONS: auth-totp
```

```
networks:
```

```
  proxy:
```

```
    external: true
```

```
volumes:
```

```
  guacamole:
```

```
    external: true
```


bar assistant

vor dem Ausrollen müssen noch folgende Volumes erstellt werde

bar-assistant bar-assistant-redis bar-assistant-meilisearch

sowie folgendes Netzwerk bar-assistant

```
version: "3"
```

```
services:
```

```
  meilisearch:
```

```
    container_name: meilisearch
```

```
    hostname: meilisearch
```

```
    image: getmeili/meilisearch:v1.1
```

```
    restart: unless-stopped
```

```
    networks:
```

```
      - bar-assistant
```

```
  volumes:
```

```
    - bar-assistant-meilisearch:/meili_data
```

```
  environment:
```

```
    - MEILI_MASTER_KEY=super-secret-password#### # password anpassen
```

```
    - MEILI_ENV=production
```

```
  redis:
```

```
    container_name: redis
```

```
    hostname: redis
```

```
    image: redis
```

```
    restart: unless-stopped
```

```
    networks:
```

```
      - bar-assistant
```

```
  volumes:
```

```
    - bar-assistant-redis:/data
```

```
  environment:
```

```
    - ALLOW_EMPTY_PASSWORD=yes
```

```
  bar-assistant:
```

```
    container_name: bar-assistant
```

```
hostname: bar-assistant
image: barassistant/server:latest
restart: unless-stopped
networks:
  - bar-assistant
volumes:
  - bar-assistant:/var/www/cocktails/storage/bar-assistant
depends_on:
  - meilisearch
  - redis
environment:
  - APP_URL=https://minibar.your-url.de/bar[ ]# url anpassen
  - LOG_CHANNEL=stderr
  - MEILISEARCH_KEY=super-secret-password[ ]# password anpassen
  - MEILISEARCH_HOST=http://meilisearch:7700
  - REDIS_HOST=redis
  - ALLOW_REGISTRATION=true
```

salt-rim:

```
container_name: salt-rim
hostname: salt-rim
image: barassistant/salt-rim:latest
restart: unless-stopped
networks:
  - bar-assistant
depends_on:
  - bar-assistant
environment:
  - API_URL=https://minibar.your-url.de/bar[ ]# url anpassen
  - MEILISEARCH_URL=https://minibar.your-url.de/search [ ]# url anpassen
  - DEFAULT_LOCALE=en-US
#   - BAR_NAME=
#   - DESCRIPTION=
```

webserver:

```
container_name: webserver
hostname: webserver
image: nginx:alpine
restart: unless-stopped
networks:
```

- bar-assistant

ports:

- 3000:3000

volumes:

- /home/your-user-folder/bar-assistant/nginx.conf:/etc/nginx/conf.d/default.conf # pfad anpassen

networks:

bar-assistant:

external: true

volumes:

bar-assistant:

external: true

bar-assistant-redis:

external: true

bar-assistant-meilisearch:

external: true

/home/your-user-folder/bar-assistant/nginx.conf

```
server {  
    listen 3000 default_server;  
    listen [::]:3000 default_server;  
    server_name _;  
  
    location = /favicon.ico { access_log off; log_not_found off; }  
    location = /robots.txt  { access_log off; log_not_found off; }  
  
    client_max_body_size 100M;  
  
    location /bar/ {  
        proxy_pass http://bar-assistant:3000/;  
    }  
  
    location /search/ {  
        proxy_pass http://meilisearch:7700/;  
    }  
  
    location / {  
        proxy_pass http://salt-rim:8080/;  
    }  
}
```

