

# Die Nextcloud docker-compose.yml

## Nun kann es endlich mit Nextcloud losgehen

Dieses Template wird immer mal aktualisiert und ist ausschließlich für eine Neuinstallation gedacht!

Der Updateprozess wird später erklärt!

## Anlegen des Programmordners

Ich habe folgende Programmstruktur bei mir angelegt: **/home/dockeruser/docker/nextcloud**

Dort werden alle benötigten Dateien zum Betrieb der Nextcloud abgelegt.

Alle nachfolgenden Schritte in dieser Anleitung beziehen sich immer auf diese Struktur!

Folgende Kommandos fügt Ihr nun als **dockeruser** aus:

```
mkdir ~/docker && mkdir ~/docker/nextcloud && cd ~/docker/nextcloud
```

Nun solltest ich im folgenden Verzeichnis sein:

```
/home/dockeruser/docker/nextcloud#
```

## Erstellen der Datei docker-compose.yml

Mit folgendem Kommando öffnet Ihr den Editor für die docker-compose.yml:

```
nano docker-compose.yml
```

Nun wird folgender Code dort hinein kopiert.

services:

db:

image: postgres:16

container\_name: nextcloud-db

restart: unless-stopped

environment:

POSTGRES\_DB: nextcloud

POSTGRES\_USER: nextcloud

POSTGRES\_PASSWORD: EIN\_SICHERES\_PASSWORT\_FESTLEGEN

volumes:

- /home/dockeruser/docker/nextcloud/db:/var/lib/postgresql/data

redis:

image: redis:7-alpine

container\_name: nextcloud-redis

restart: unless-stopped

command: ["redis-server", "--appendonly", "yes"]

volumes:

- /home/dockeruser/docker/nextcloud/redis:/data

app:

image: nextcloud:31-fpm

container\_name: nextcloud-app

restart: unless-stopped

depends\_on:

- db

- redis

environment:

POSTGRES\_HOST: db

POSTGRES\_DB: nextcloud

POSTGRES\_USER: nextcloud

POSTGRES\_PASSWORD: EIN\_SICHERES\_PASSWORT\_FESTLEGEN

REDIS\_HOST: redis

volumes:

- /home/dockeruser/docker/nextcloud/nextcloud:/var/www/html

- /home/dockeruser/docker/nextcloud/config:/var/www/html/config

web:

image: nginx:alpine

container\_name: nextcloud-web

```

restart: unless-stopped
depends_on:
  - app
ports:
  - "8088:80"
volumes:
  - /home/dockeruser/docker/nextcloud/nextcloud:/var/www/html:ro
  - /home/dockeruser/docker/nextcloud/config:/var/www/html/config:ro
  -
/home/dockeruser/docker/nextcloud/nginx/default.conf:/etc/nginx/conf.d/default.conf:ro

collabora:
  image: collabora/code:latest
  container_name: collabora
  restart: unless-stopped
  environment:
    - domain=office\\.mein\\.dyndns\\.local
    - username=EINEN_BENUTZERNAMEN_FESTLEGEN
    - password=EIN_SICHERES_PASSWORT_FESTLEGEN
    - extra_params=-o:ssl.enable=false --o:ssl.termination=true
  ports:
    - "9980:9980"

cron:
  image: nextcloud:31-fpm
  container_name: nextcloud-cron
  restart: unless-stopped
  depends_on:
    - db
    - redis
  volumes:
    - /home/dockeruser/docker/nextcloud/nextcloud:/var/www/html
    - /home/dockeruser/docker/nextcloud/config:/var/www/html/config
  entrypoint: /cron.sh

```

**WICHTIG: Handelt hier vorsichtig und mit wirklich starken Passwörtern!**

Im nano ändert Ihr vor den Speichern den Inhalt der Variablen folgender Zeilen **9, 32, 56, 58, 59**!

# Der Reverse-Proxy (traefik)

Um die Seite im Internet bereit zu stellen benutzen wir einen Reverse-Proxy der den Nextcloud-Dienst ausschließlich über TLS-Verschlüsselung anbietet (https). Im Heimsetzerk ist die Nextcloud über http erreichbar, wird aber darüber nicht genutzt.

Ich betreibe einen Traefik Dienst im Docker der als ein HTTP-[Reverse-Proxy](#) und Load Balancer dient, der die Bereitstellung von Microservices erleichtert. Dieser läuft auf einer anderen Maschine, da ich dort viele Microservices laufen lasse (unter anderen dieses Wiki) und ich will die Nextcloud-Instanz nicht auch dort noch laufen lassen, aus Performanz gründen.

**Zeile 56** dient zur Definition **FQDNS** für Collabora. Die \\. müssen drin bleiben!

## Die initiale nginx default.conf

Bitte führe folgenden Befehl in der Konsole aus:

```
cat > /home/dockeruser/docker/nextcloud/nginx/default.conf <<'EOF'
upstream php-handler {
    server app:9000;
}

server {
    listen 80;
    server_name _;

    include /etc/nginx/mime.types;

    types { text/javascript mjs; }

    client_max_body_size 10G;
    fastcgi_buffers 64 4K;

    root /var/www/html;

    add_header Referrer-Policy "no-referrer" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Permitted-Cross-Domain-Policies "none" always;
    add_header X-Robots-Tag "noindex, nofollow" always;
    add_header X-XSS-Protection "1; mode=block" always;
```

```
add_header Strict-Transport-Security "max-age=15552000" always;

index index.php index.html /index.php$request_uri;

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}

location = /.well-known/carddav { return 301 /remote.php/dav/; }
location = /.well-known/caldav { return 301 /remote.php/dav/; }

location / {
    try_files $uri $uri/ /index.php$request_uri;
}

location ~ /\.php(?:$|/) {
    rewrite
    ^/(?!index|remote|public|cron|core\ajax\update|status|ocs\v[12]|updater\./+|ocs-
provider\./+|\./richdocumentscode/proxy) /index.php$request_uri;

    fastcgi_split_path_info ^(.+?\php)(/.*)$;
    set $path_info $fastcgi_path_info;

    try_files $fastcgi_script_name =404;

    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $path_info;
    fastcgi_param HTTPS on;
    fastcgi_param modHeadersAvailable true;
    fastcgi_param front_controller_active true;

    fastcgi_pass php-handler;
    fastcgi_intercept_errors on;
    fastcgi_request_buffering off;
}

location ~ \.(?:css|js|mjs|svg|gif|png|jpg|ico|wasm|tflite|map)$ {
```

```
try_files $uri /index.php$request_uri;
add_header Cache-Control "public, max-age=15778463, immutable";
access_log off;
expires 6M;
}

location ~ \.(?:woff2?|otf|ttf)$ {
    try_files $uri /index.php$request_uri;
    expires 7d;
    access_log off;
}

location /remote {
    return 301 /remote.php$request_uri;
}

location /updater {
    try_files $uri/ =404;
    index index.php;
}
}
EOF
```

---

Revision #18

Created 2026-02-14 12:18:09 UTC by sylvio

Updated 2026-02-15 13:07:49 UTC by sylvio