



Nextcloud über Docker

Inhaltsverzeichnis

Zusammenfassung	1
Voraussetzung	1
Hinweis	1
Schritt für Schritt	2
Nextcloud und DB über Docker	2
Nextcloud-Container automatisch starten	4
SSL-Verbindung mit Reverse-Proxy	7
Reverse-Proxy automatisch starten	11
Abschluss	14

Zusammenfassung

In diesem Dokument erfahren Sie, wie Sie eine Nextcloud in einem Container installieren.

Voraussetzung

In diesem Dokument wird davon ausgegangen, dass

- Sie bereits ein Debian Linux ohne Desktop-Environment installiert haben.
- Sie bereits Docker installiert haben.

Hinweis

- Bei einem Linux-Befehle bedeutet das `\`, dass der Befehl in der nächsten Zeile fortgesetzt wird. Dies dient der Lesbarkeit. Dieser ist nur zu schreiben, wenn der Befehl kopiert wird. Befehle können auch ohne `\` in einer Zeile angegeben werden.

Schritt für Schritt

Nextcloud und DB über Docker

1. Installieren Sie das Paket docker-compose.

Erstellung der Ordner

```
mkdir -p /srv/docker/nextcloud/data  
mkdir -p /srv/docker/nextcloud/db  
cd /srv/docker/nextcloud/
```

- Erstellen Sie die Datei `docker-compose.yml` und tragen Sie folgenden Inhalt ein:



YAML-Dateien benötigen eine bestimmte Formatierung. Die Einrückungen müssen mit 2x **[Leertaste]** eingerückt werden. Wenn es später zu Fehlermeldungen während des Compose kommt, sind die Einrückungen nicht korrekt oder es fehlen bestimmte Leertasten zwischen dem **[:]** und Übergabewert.

Mehr: [YAML Grundlagen](#) | [Edi's Techlab](#)



docker-compose.yml

```
version: '3'

volumes:
  data:
  config:
  db:

services:
  nextcloud-db:
    image: mariadb
    container_name: nextcloud-db
    restart: always
    volumes:
      - ./db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=Muster! ①
      - MYSQL_PASSWORD=Muster! ②
      - MYSQL_DATABASE=nextcloud ③
      - MYSQL_USER=nextcloud ④
  nextcloud:
    image: nextcloud
    container_name: nextcloud
    ports:
      - 80:80
    depends_on:
      - nextcloud-db
    volumes:
      - ./data:/var/www/html/data
```

- ① Geben Sie Ihr Root-Passwort für die Datenbank ein.
- ② Geben Sie ein Benutzerpasswort ein. Diess wird später für die Erstkonfiguration benötigt.
- ③ Geben Sie eine Datenbank an. Diese wird später für die Erstkonfiguration benötigt.
- ④ Geben Sie einen Benutzernamen ein. Dieser wird später für die Erstkonfiguration benötigt.

Container starten

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



Nextcloud-Container automatisch starten

- Erstellen Sie die Datei `start.sh` und tragen Sie folgenden Inhalt ein:

start.sh

```
#!/bin/sh

RC=0
cd /srv/docker/nextcloud
/usr/bin/docker-compose up -d || RC=1
exit $RC
```

- Erstellen Sie die Datei `stop.sh` und tragen Sie folgenden Inhalt ein:

stop.sh

```
#!/bin/sh

RC=0
cd /srv/docker/nextcloud
/usr/bin/docker-compose down || RC=1
exit $RC
```

Berechtigung ändern

```
chmod 755 start.sh
chmod 755 stop.sh
```

- Erstellen Sie die Datei `/etc/systemd/system/nextcloud.service` und tragen Sie folgenden Inhalt ein:



nextcloud.service

```
[Unit]
Description=Docker - Nextcloud Container
Requires=docker.service
After=docker.service

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/srv/docker/nextcloud/start.sh
ExecStop=/srv/docker/nextcloud/stop.sh

[Install]
WantedBy=default.target
```

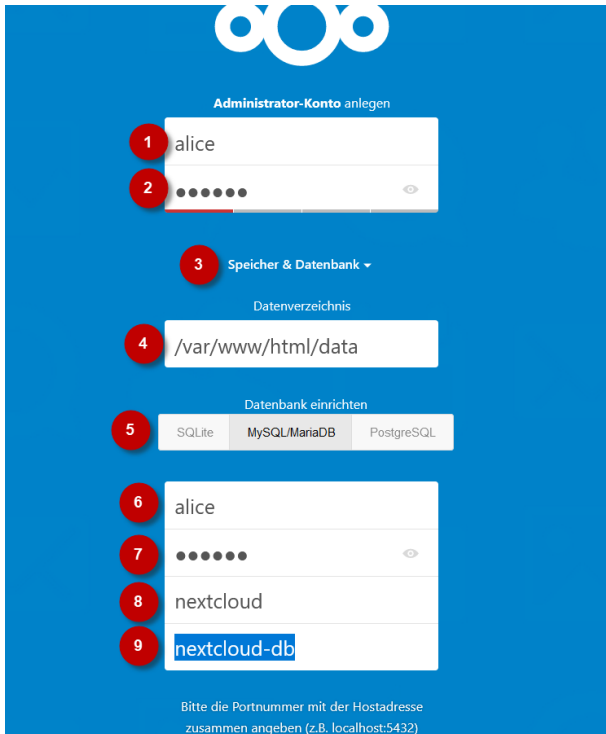
Service aktivieren

```
systemctl enable nextcloud.service
```

Service starten

```
systemctl start nextcloud.service
```

- Starten Sie Ihren Browser und geben Sie als URL http://Ihre_IP-Adresse ein.



1. Geben Sie einen **Administratorname** ein.
2. Geben Sie ein **Passwort** ein.
3. Klicken Sie auf [**Speicher & Datenbank**]
4. Geben Sie den Pfad **/var/www/html/data** ein.
5. Wählen Sie [**MySQL/MariaDB**] aus.
6. Geben Sie den **Benutzer** der Datenbank ein.
7. Geben Sie das **Passwort** der Datenbank ein.
8. Geben Sie die **Datenbank** ein.
9. Geben Sie den Container **nextcloud-db** ein.



Daten aus der **/srv/docker/nexcloud/docker-compose.yml** .

6 = MYSQL_USER

7 = MYSQL_PASSWORD

8 = MYSQL_DATABASE

SSL-Verbindung mit Reverse-Proxy



Selbst signierte Zertifikate taugen nur für den Test- beziehungsweise privaten Betrieb. Für den Produktivbetrieb sollte man in Erwägung ziehen, offizielle Zertifikate, zum Beispiel von Let's Encrypt, einzusetzen.

Daten der Nextcloud löschen

```
systemctl stop nextcloud.service  
rm -rf /srv/docker/nextcloud/data/* /srv/docker/nextcloud/db/*
```

Verzeichnis erstellen

```
mkdir -p /srv/docker/nginx-proxy/ssl  
mkdir -p /srv/docker/nginx-proxy/conf.d  
cd /srv/docker/nginx-proxy/ssl
```

Erstellung eines privaten Schlüssels

```
openssl genrsa -out proxy.key 4096
```

Erstellung des Certificate Signing Request

```
openssl req -new -key proxy.key -out proxy.csr
```



proxy.csr

```
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:BE
Locality Name (eg, city) []:Berlin
Organization Name (eg, company) [Internet Widgits Pty Ltd]:JIKU
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:debian
Email Address []:
Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:
An optional company name []:
```



Password leer lassen, sonst muss man bei jedem Start des Containers das Passwort in die Console eingeben.

Erstellung eines Self-Sign-Certificate

```
openssl x509 -req -days 365 -in proxy.csr -signkey proxy.key -out proxy.crt
```

1. Wechseln Sie das Verzeichnis nach [/srv/docker/nginx-proxy](#)
2. Erstellen Sie die Datei [docker-compose.yml](#) und tragen Sie folgenden Inhalt ein:



docker-compose.yml

```
version: '2'

volumes:
  docker.sock:
  ssl:
  conf.d:

services:
  nginx-proxy:
    image: jwilder/nginx-proxy
    container_name: nginx-proxy
    ports:
      - "443:443"
    volumes:
      - /var/run/docker.sock:/tmp/docker.sock:ro
      - ./ssl:/etc/nginx/certs:ro
      - ./conf.d:/etc/nginx/conf.d
  whoami:
    image: jwilder/whoami
    container_name: whoami
    environment:
      - VIRTUAL_HOST=whoami.local
```



1. Wechseln Sie das Verzeichnis nach `/srv/docker/nginx-proxy/conf.d/`
2. Erstellen Sie die Datei `nextcloud.conf` und tragen Sie folgenden Inhalt ein:

nextcloud.conf

```
upstream backend {  
    server IP-Adresse:80;  
}  
  
server {  
    listen 443 ssl;  
    ssl_certificate /etc/nginx/certs/proxy.crt;  
    ssl_certificate_key /etc/nginx/certs/proxy.key;  
    location / {  
        proxy_pass http://IP-Adresse;  
    }  
}
```

- Wechseln Sie das Verzeichnis nach `/srv/docker/nginx-proxy/`

Container starten

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



Reverse-Proxy automatisch starten

- Erstellen Sie die Datei `start.sh` und tragen Sie folgenden Inhalt ein:

start.sh

```
#!/bin/sh

RC=0
cd /srv/docker/nginx-proxy
/usr/bin/docker-compose up -d || RC=1
exit $RC
```

- Erstellen Sie die Datei `stop.sh` und tragen Sie folgenden Inhalt ein:

stop.sh

```
#!/bin/sh

RC=0
cd /srv/docker/nginx-proxy
/usr/bin/docker-compose down || RC=1
exit $RC
```

Berechtigung ändern

```
chmod 755 start.sh
chmod 755 stop.sh
```



- Erstellen Sie die Datei `/etc/systemd/system/nginx-proxy.service` und tragen Sie folgenden Inhalt ein:

nextcloud.service

```
[Unit]
Description=Docker - Nginx-Proxy Container
Requires=docker.service
After=docker.service

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/srv/docker/nginx-proxy/start.sh
ExecStop=/srv/docker/nginx-proxy/stop.sh

[Install]
WantedBy=default.target
```

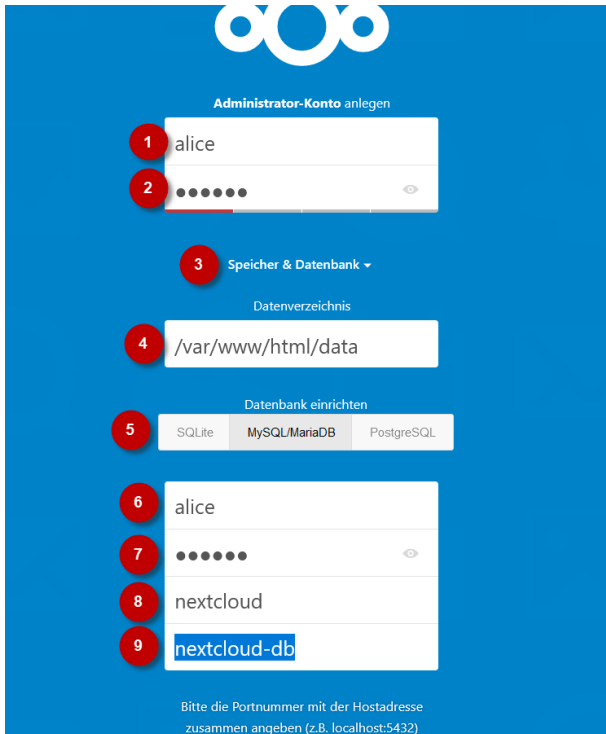
Service aktivieren

```
systemctl enable nginx-proxy.service
```

Service komplett starten

```
systemctl start nginx-proxy.service
```

1. Starten Sie das Debian Linux neu.
2. Starten Sie Ihren Browser und geben Sie als URL https://Ihre_IP-Adresse ein.



1. Geben Sie einen **Administratortnamen** ein.
2. Geben Sie ein **Passwort** ein.
3. Klicken Sie auf [**Speicher & Datenbank**]
4. Geben Sie den Pfad **/var/www/html/data** ein.
5. Wählen Sie [**MySQL/MariaDB**] aus.
6. Geben Sie den **Benutzer** der Datenbank ein.
7. Geben Sie das **Passwort** der Datenbank ein.
8. Geben Sie die **Datenbank** ein.
9. Geben Sie den Container **nextcloud-db** ein.



Daten aus der **/srv/docker/nexcloud/docker-compose.yml** .

6 = MYSQL_USER

7 = MYSQL_PASSWORD

8 = MYSQL_DATABASE



Designing Education
Connecting People

Abschluss

Sie haben erfolgreich eine Nextcloud als Docker-Container implementiert und via HTTPS abgesichert. Sie können nun mit der nächsten Übung fortfahren.



Erstellen Sie einen Prüfpunkt und benennen Sie ihn, bevor Sie mit der nächsten Übung beginnen.