HowTo install Onlyoffice in Nextcloud (Teil2)?

Nextcloud: Online-Office mit ONLYOFFICE (mit eigener Subdomain)

9. November 2018JanHome-Server, 136

Nextcloud bietet neben dem Speichern von Dateien auch noch erweiterte Funktionen wie z.B. die Verwaltung von Kontakten und Kalendern. Ebenso sind Online-Office-Funktionalitäten mit der eigenen Cloud-Lösung möglich: Hier gibt es die Alternativen ONLYOFFICE und Collabora. Für welche Lösung man sich hier entscheidet, ist zunächst einmal Geschmackssache.

Zum Thema ONLYOFFICE gab es bereits einen Artikel in diesem Blog. Die hier gezeigte Lösung hatte allerdings einen entscheidenden Nachteil: Die Office-Funktionalitäten waren nur verfügbar, wenn man über das lokale Netzwerk (LAN) auf die eigene Cloud zugegriffen hat. Das Bearbeiten von Dokumenten über das Internet war auf diese Art und Weise nicht möglich, da beim Bearbeiten von Dokumenten immer direkt auf die IP zugegriffen wurde, auf der ONLYOFFICE installiert wurde. Diese LAN-IP war über das Internet natürlich nicht erreichbar.

In diesem Artikel soll es daher wieder mal um die Einrichtung von ONLYOFFICE mittels Docker unter Nextcloud gehen. Dieses Mal sollen jedoch keine Einschränkungen für die Benutzung des Online-Office gelten: Somit ist dann der Zugriff sowohl über das lokale Netzwerk, also auch über das Internet möglich.

Als Grundlage dient wie immer der Artikel Nextcloud auf Ubuntu Server 18.04 LTS mit nginx, MariaDB, PHP, Let’s Encrypt, Redis und Fail2ban.

Update-Historie (letztes Update 05.09.2019)

Inhalt [hide]

Zweite (Sub-)Domain für ONLYOFFICE

Ich gehe in diesem Artikel davon aus, dass der eigene Server, auf dem Nextcloud bereits läuft, über die DynDNS-Domain **meindomain.de** erreichbar ist. DynDNS sorgt hier dafür, dass die öffentliche IP-Adresse des Routers auf eine Domain gemappt wird. Im Router wird die Domain dazu in den DynDNS-Einstellungen hinterlegt. Leider unterscheidet sich das Vorgehen von Router zu Router, daher würde eine detaillierte Anleitung für verschiedene Router-Modelle dem Umfang des Artikels sprengen.

Voraussetzung für ONLYOFFICE ist nun eine **zweite (Sub-)Domain**, über die der Dokumenten-Server später erreichbar sein wird. Als Beispiel nehme ich hier einfach die Domain **onlyoffice.meinedomain.de**. Diese zweite Domain muss dabei ebenfalls auf die öffentliche IP-Adresse des Routers „zeigen“, ist sozusagen auch eine zweite (oder besser gesagt alternative) DynDNS-Domain.

Mehrere DynDNS-Domains mittels CNAME

Und genau das ist hier das Problem. In fast allen Router-Modellen kann nur ein DynDNS-Anschluss mit nur einer Domain festgelegt werden. Die Lösung für dieses Problem stellt ein sog. **CNAME-Eintrag** für eine zweite (Sub-)Domain dar. Mittels CNAME wird einfach ausgedrückt ein alternativer Name für eine Domain hinterlegt: Der CNAME-Eintrag der Office-Domain muss daher die bereits vorhandene DynDNS-Domain sein.

Wo kann man diesen CNAME-Eintrag nun hinterlegen? Dies passiert normalerweise in der Domain-Verwaltung des Webspace-Providers. Auch hier unterscheidet sich das Vorgehen von Anbieter zu Anbieter, gute Erfahrungen konnte ich hier allerdings mit All-Inkl.com (Affiliate-Link) machen.

Hier loggt man sich zunächst im Kundenportal (KAS) ein und legt eine neue Subdomain an. Dazu findet man im Menü den Eintrag Subdomain. Über den Punkt Neue Subdomain anlegen kann hier eine neue Subdomain beantragt werden. Hier kann man alle Einstellungen einfach übernehmen, da diese Domain später nicht auf den Webspace, sondern die öffentliche IP-Adresse des eigenen Routers zeigen wird.

All-Inkl.com: Neue Subdomain anlegen

Um nun noch den CNAME-Eintrag für die neue Subdomain hinzuzufügen, muss man die Domain-Einstellungen unter Tools > DNS-Einstellungen bearbeiten. Hier sieht man dann die eigene Domain, über die auch der DynDNS-Anschluss läuft (meinedomain.de). Mit einem Klick auf Bearbeiten kann diese Domain modifiziert werden. Hier klickt man dann auf den Punkt neuen DNS Eintrag erstellen. Folgende Einstellungen sind hier vorzunehmen:

- **Name:** Der Name der Subdomain (hier also onlyoffice.meinedomain.de).
- **Typ/Prio.:** CNAME

- **Data:** Hier wird die eigentliche DynDNS-Domain eingetragen (also meinedomain.de).

All-Inkl.com: CNAME-Eintrag für Subdomain erstellen

Nach einem Klick auf Speichern werden die Änderungen übernommen.

SSL-Zertifikat für die neue Domain hinzufügen

Damit die Verbindung zu ONLYOFFICE stets verschlüsselt, also über HTTPS abläuft, muss für die neue Subdomain noch ein SSL-Zertifikat über Let's Encrypt erzeugt werden. Dazu wird das bereits für Nextcloud verwendete Zertifikat modifiziert, indem wir dieses Zertifikat um eine weitere (Sub-)Domain erweitern. **Hierfür sind keine Änderungen an den virtuellen Hosts von nginx erforderlich**. Ebenfalls müssen bereits vorhandene Diffie-Hellman-Parameter (siehe Beschreibung) nicht neu erzeugt werden und können aus der bestehenden Installation übernommen werden.

Certbot

Wenn für die Generierung der Zertifikate Certbot zum Einsatz kommt, dann reicht hier folgender Befehl auf der Kommandozeile, um neue Zertifikate für beide (Sub-)Domains auszustellen:

```
certbot certonly --webroot -w /var/www/letsencrypt -d meinedomain.de -d onlyoffice.m
einedomain.de --rsa-key-size 4096
```

Wichtig ist hier die Angabe **aller** (Sub-)Domains mit dem Parameter -d. Nach diesem Befehl werden die Zertifikats-Dateien unter /etc/letsencrypt/live/meinedaomin.de erneuert. Nicht wundern, hier wird nicht für jede (Sub-)Domain ein Zertifikat erzeugt, sondern nur **ein Zertifikat**, welches für mehrere Domains ausgestellt ist.

acme.sh

Falls acme.sh für die Generierung der Zertifikate verwendet wird, sollten zunächst die Grundlagen zu acme.sh aus dem Artikel Let's Encrypt Zertifikate mit acme.sh und nginx entnommen werden.

Der konkrete Befehl zum Erzeuigen der Zertifikate sieht in diesem Fall dann so aus:

```
acme.sh --issue -d meinedomain.de -d onlyoffice.meinedomain.de --keylength 4096 -w /
var/www/letsencrypt --key-file /etc/letsencrypt/meinedomain.de/key.pem --ca-file /et
c/letsencrypt/meinedomain.de/ca.pem --cert-file /etc/letsencrypt/meinedomain.de/cert
.pem --fullchain-file /etc/letsencrypt/meinedomain.de/fullchain.pem --reloadcmd "sud
o /bin/systemctl reload nginx.service"
```

Hier werden dann ebenfalls **alle** (Sub-)Domains mittels des Parameters -d angegeben und in einem einzigen Zertifikat „verpackt“.

Auslagern der SSL-Einstellungen aus dem virtuellen Host für Nextcloud

Für die neue Domain muss nun natürlich ein neuer virtueller Host angelegt werden. Dieser zielt wieder auf SSL ab, daher sind hier auch sämtliche SSL-Einstellungen wie ssl_certificate, ssl_protocols, ssl_ciphers, etc. notwendig. Damit wir diese nicht doppelt pflegen müssen (im Gateway-Host und im virtuellen Host für ONLYOFFICE), lagern wir diese Anweisungen für SSL einfach aus.

Vorher sollten alle virtuellen Hosts gesichert werden, falls beim Umzug etwas schief gehen sollte.

Dazu erstellen wir zunächst einen neuen Ordner, in dem die SSL-Einstellungen gespeichert werden sollen:

```
mkdir /etc/nginx/snippets
```

In diesem Ordner erstellen wir anschließend eine neue Datei speziell für allgemeine SSL-Einstellungen, die für alle (Sub-)Domains gelten sollen.

Hinweis: Diese ausgelagerten Einstellungen dürfen nicht im selben Ordner wie die restlichen vHosts (/etc/nginx/conf.d) gespeichert werden, da es ansonsten Probleme mit dem Setzen der HTTP-Header gibt.

In diese Datei werden nun die Anweisungen übernommen, die für die SSL- und Header-Einstellungen zuständig sind. In diesem konkreten Beispiel (siehe hier) sind die Zeilen 32-95 im **Gateway-Host** betroffen. Dieser Block wird nun aus dem Gateway-Host kopiert und in die ssl.conf eingefügt.

```
nano /etc/nginx/snippets/ssl.conf
```

In diesem Beispiel sind es die folgenden Inhalte.

```
# Certificates used
ssl_certificate /etc/letsencrypt/live/meinedomain.de/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/meinedomain.de/privkey.pem;

# Not using TLSv1 will break:
# Android <= 4.4.40
# IE <= 10
# IE mobile <=10
# Removing TLSv1.1 breaks nothing else!
# TLSv1.3 is not supported by most clients, but it should be enabled.
ssl_protocols TLSv1.2 TLSv1.3;

# Cipher suite from https://cipherli.st/
# Max. security, but lower compatibility
ssl_ciphers 'ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-
AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384';

# Cipher suite from https://wiki.mozilla.org/Security/Server_Side_TLS
#ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RS
A-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-
AES128-SHA256:ECDHE-RSA-AES128-SHA256';
```

```
# (Modern) cipher suite from https://mozilla.github.io/server-side-tls/ssl-config-
generator/
#ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RS
A-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-
AES128-SHA256:ECDHE-RSA-AES128-SHA256';

# Diffie-Hellman parameter for DHE ciphersuites, recommended 4096 bits
ssl_dhparam /etc/nginx/ssl/dhparams.pem;

# Use multiple curves.
# secp521r1: Not supported by Chrome
# secp384r1: Not supported by Android (DAVdroid)
ssl_ecdh_curve secp521r1:secp384r1:prime256v1;

# Server should determine the ciphers, not the client
ssl_prefer_server_ciphers on;

# OCSP Stapling
# fetch OCSP records from URL in ssl_certificate and cache them
ssl_stapling on;
ssl_stapling_verify on;

# This should be chain.pem
# See here: https://certbot.eff.org/docs/using.html
ssl_trusted_certificate /etc/letsencrypt/live/meinedomain.de/chain.pem;

resolver 192.168.178.1;

# SSL session handling
ssl_session_timeout 24h;
ssl_session_cache shared:SSL:50m;
ssl_session_tickets off;

#
# Add headers to serve security related headers
#
# HSTS (ngx_http_headers_module is required)
# In order to be recoginzed by SSL test, there must be an index.hmtl in the server's
 root
add_header Strict-Transport-Security "max-
age=63072000; includeSubdomains; preload;";
add_header X-Content-Type-Options "nosniff";
add_header Referrer-Policy "no-referrer";
add_header X-XSS-Protection "1; mode=block";
add_header X-Robots-Tag none;
add_header X-Download-Options noopen;
add_header X-Permitted-Cross-Domain-Policies none;

# Remove X-Powered-By, which is an information leak
fastcgi_hide_header X-Powered-By;
```

Im nächsten Schritt werden nun diese SSL-Einstellungen **aus dem Gateway-Host entfernt**. Beispielweise also wieder die Zeilen 32-95. Damit die SSL-Einstellungen nun auch für den Gateway-Host gelten, wird nun statt dem entfernten Block einfach folgende Zeile mit übernommen:

```
include /etc/nginx/snippets/ssl.conf;
```

Wichtig ist hier v.a., dass die Anweisungen nicht doppelt aufgeführt sind (einmal im Gateway-Host direkt und einmal durch den Include). Daher müsen die eigentlichen SSL-Anweisungen auch aus dem Gateway-Host entfernt werden.

Anschließend starten wir den Webserver neu:

```
service nginx restart
```

Nun sollte man zunächst einmal testen, ob alle vorhandenen Web-Anwendungen (wie z.B. Nextcloud) noch wie erwartet funktionieren. Erst wenn diese Überprüfung erfolgreich war, sollte man hier weitermachen.

Virtuellen Host für ONLYOFFICE hinzufügen

Alles funktioniert noch? Gut! Dann weiter...

Im nächsten Schritt wird für die neue Subdomain ein eigener vHost hinzugefügt:

Shell

1 nano

/etc/nginx/conf.d/onlyoffice
.meinedomain.de.conf

Der Inhalt ist hier recht übersichtlich:

Shell

```
1  server {
2      listen 443 ssl http2;
3      server_name onlyoffice
.meinedomain.de;
4
5
       # Include SSL configuration
6
       include /etc/nginx/snippets/ssl.conf;
7
```

```
8
9                       #
10                      # Configuration for OnlyOffice
11                      #
12                      location / {
13                              proxy_pass https://
192.168.178.60:4433;
14
                                proxy_redirect off;
15
                                proxy_set_header Host $host;
16
                                proxy_set_header X-Real-IP
17                      $remote_addr;
18                              proxy_set_header X-
Forwarded-For
19                      $proxy_add_x_forwarded_for;
20                              proxy_set_header X-
Forwarded-Host $server_name;

                                proxy_set_header X-
Forwarded-Proto $scheme;

                        }

                }
```

Die SSL-Einstellungen haben wird ja zuvor ausgelagert, so dass diese nun in diesem zweiten vHost einfach wiederverwendet werden können.

Wichtig bei der Angabe des proxy_pass:

- Für die Verbindung zur Maschine, auf der später ONLYOFFICE laufen wird, nutzen wir ebenfalls **HTTPS**. Wenn dies der gleiche Rechner ist, auf dem auch Nextcloud schon läuft, könnte man hier auch einfach auf HTTP setzen, da die Kommunikation dann nur Rechner-intern abläuft (dazu später mehr).
- Die IP ist hier die **lokale IP-Adresse** des Rechners, auf dem ONLYOFFICE später laufen wird. Dies kann also die IP der Maschine sein, auf der auch der Webserver (und z.B. Nextcloud) läuft, oder auch eine andere Maschine im LAN.
- Als Port habe ich hier 4433 angegeben. Dies ist besonders wichtig, wenn ONLYOFFICE auf dem gleichen Rechner installiert wird, auf dem auch der Webserver/Nextcloud läuft. Der

Standard-Port für HTTPS 443 wäre hier schon (durch den Gateway-Host) belegt, also muss man auf einen anderen Port ausweichen.

Installation und Einrichtung ONLYOFFICE

Nachdem nun sämtliche Voraussetzungen erfüllt sind, kann es an die Installation von ONLYOFFICE gehen. Ich bevorzuge hier den Weg über Docker, da eine komplett manuelle Installation sehr aufwendig ist. Hier ist es auch egal, auf welcher Maschine ONLYOFFICE installiert wird. Entweder man installiert das Online-Office direkt auf dem Rechner, auf dem auch Nextcloud bereits läuft, oder man nimmt eine zweite Maschine (oder VM), auf der dann nur ONLYOFFICE betrieben wird.

Docker sollte auf jeden Fall bereits installiert sein. Fall nicht, sollte dieser Schritt wie im Artikel Docker auf Ubuntu Server nun nachgeholt werden.

Alle folgenden Schritte müssen auf der Maschine ausgeführt werden, auf der ONLYOFFICE installiert werden soll.

Vorbereitungen für die Installation

Die Verbindung zu ONLYOFFICE sollte aus Sicherheitsgründen immer verschlüsselt, also über HTTPS ablaufen. Durch den vHost für ONLYOFFICE ist die Verbindung über das Internet auch bereits verschlüsselt. Bleibt nur der Verbindungsweg vom Webserver auf die Maschine, auf der das Online-Office später laufen wird (also die Verbindung im lokalen Netzwerk). Diese Verbindung sollte ebenfalls verschlüsselt werden, damit auch LAN-intern stets sichere Verbindungen aufgebaut wird.

Wenn ONLYOFFICE auf dem gleichen Rechner gehostet wird, auf dem schon Nextcloud und der Webserver laufen, ist dieser Schritt eigentlich optional, da hier nur eine Rechner-interne Verbindung zustande kommt. Trotzdem empfehle ich diese Schritte auszuführen, besonders wenn die Office-Lösung und der Webserver/Nextcloud auf unterschiedlichen Maschinen installiert werden.

Da wir hier ja nur mit einer IP-Adresse arbeiten (die beim proxy_pass angegeben wird), können wir hierfür kein Let’s Encrypt Zertifikat erstellen, da diese Zertifikate nicht auf IP-Adressen, sondern nur auf Domains ausgestellt werden können. Daher greifen wir hier auf ein selbst **signiertes Zertifikat** zurück.

Dieses wird mit folgenden Befehlen erzeugt:

Shell

```
1  mkdir -p
   /app/onlyoffice/
2  DocumentServer/data/certs

3  cd
   /app/onlyoffice/
4  DocumentServer/data/certs

5  openssl genrsa -out onlyoffice.key 4096

6  openssl req -new -key onlyoffice.key
   -out onlyoffice.csr
7
   openssl x509 -req -days 3650 -in
8  onlyoffice.csr -signkey onlyoffice.key
   -out onlyoffice.crt
9
   openssl dhparam -out dhparam.pem 4096
10
   chmod 400 onlyoffice.key

   chmod 400 onlyoffice.crt

   chmod 400 onlyoffice.csr

   chmod 400 dhparam.pem
```

- Beim Befehl openssl req -new -key onlyoffice.key -out onlyoffice.csr wird man nach dem „Common Name“ gefragt (Common Name (e.g. server FQDN or YOUR name)). Hier ist einfach die IP des lokalen Systems anzugeben (in diesem Fall 192.168.178.60). Ebenso kann man ein „challenge password“ angeben. Dieses kann man einfach leer lassen (einfach mit Enter bestätigen).
- **Achtung:** Die Erzeugung der sog. Diffie-Hellmann-Parameter mit einer Länge von 4096 Bit (openssl dhparam -out dhparam.pem 4096) kann u.U. sehr lange dauern. Auf schwacher Hardware kann das schon einmal mehrere Stunden in Anspruch nehmen. Wer nicht so lange warten möchte, kann die Schlüssel-Länge auf 2048 Bit reduzieren (openssl dhparam -out dhparam.pem 2048).

- Das Zertifikat hat eine Gültigkeit von 3650 Tagen (10 Jahre). Hier kann bei Bedarf auch eine andere Gültigkeitsdauer angegeben werden.

Installation ONLYOFFICE

Die Installation beschränkt sich dann dank Docker auf einen einzigen Befehl:

Shell

```
1  docker run --name=ONLYOFFICEDOCKER -i -t -d -p 4433:443 -e JWT_ENABLED='true' -e JWT_SECRET='geheimes-secret' --restart=always -v /app/onlyoffice/DocumentServer/data:/var/www/onlyoffice/Data onlyoffice/documentserver
```

Hier gilt es folgende Punkte zu beachten:

- Den Container nennen wir hier ONLYOFFICEDOCKER, damit dieser später über diesen Namen und nicht über eine kryptische ID angesprochen werden kann.
- Da bereits der Webserver auf Port 443 lauscht (wenn ONLYOFFICE auf dem gleichen Rechner gestartet wird), müssen wird hier auf einen alternativen Port ausweichen. Der Parameter -p 4433:443 sorgt hier dafür, dass der Port 4433 des Docker-Hosts auf den Port 443 innerhalb des Docker-Containers gemappt wird. Wichtig ist hier, dass dieser Port (4433) mit demjenigen übereinstimmt, der beim proxy_pass im vHost für ONLYOFFICE angegeben wurde.
- Die nächsten zwei Parameter (-e JWT_ENABLED=’true‘ -e JWT_SECRET=’geheimes-secret‘) erhöhen die Sicherheit, da zur Kommunikation mit dem Container ein sog. JSON Web Token benötigt wird. Im Grunde genommen handelt es sich dabei um ein Passwort (geheimes-secret – **hier sollte man natürlich ein eigenes Passwort wählen**), welches später in Nextcloud hinterlegt werden muss, damit die Verbindung zu ONLYOFFICE aufgebaut werden kann. Dies verhindert, dass der ONLYOFFICE-Container „unbemerkt“ von anderen Verbindungen genutzt werden kann.
- Mit —restart=always wird der Container bei jedem Systemstart automatisch gestartet.
- Mit dem letzten Parameter (-v /app/onlyoffice/DocumentServer/data:/var/www/onlyoffice/Data onlyoffice/documentserver) wird ein sog. Volume definiert: Alle Dateien, die im Verzeichnis /app/onlyoffice/DocumentServer/data des Hosts liegen, werden innerhalb des Containers im Verzeichnis /var/www/onlyoffice/Data onlyoffice/documentserver bereitgestellt. Diese Funktion wird benötigt, damit der Container das zuvor erzeugte Zertifikat finden und nutzen kann.

Nun wird das entsprechende Docker-Image heruntergeladen und auch gleich gestartet.

Installation von ONLYOFFICE mittels Docker

Nach dem Starten des Containers läuft ONLYOFFICE bereits. Wenn man nun im Browser die Domain onlyoffice.meinedomain.de oder die IP-Adresse der jeweiligen Maschine (dann aber mit dem alternativen Port 4433) aufruft, sollte folgende Seite erscheinen:

ONLYOFFICE läuft

Einbinden von ONLYOFFICE in Nextcloud

Nachdem ONLYOFFICE läuft, geht es nun mit der eigenen Nextcloud weiter, damit hier Online-Office-Funktionalitäten hinzugefügt werden können.

Die passende App findet man dabei im Nextcloud App Store in der Kategorie Büro & Text:

ONLYOFFICE im Nextcloud App-Store

Wenn die App aktiviert wurde, können die Einstellungen nun in der Admin-Oberfläche von Nextcloud unter dem Punkt ONLYOFFICE angepasst werden (damit alle Einstellungen angezeigt werden, muss man auf den Punkt Erweiterte Servereinstellungen klicken).

Nextcloud: ONLYOFFICE-Einstellungen in der Admin-Oberfläche

- Serviceadresse der Dokumentbearbeitung: Hier wird die Subdomain angegeben, die wir extra für ONLYOFFICE angelegt habe, in diesem Beispiel also https://onlyoffice.meinedomain.de.
- Geheimer Schlüssel (freilassen, um zu deaktivieren): Hier ist das JWT-Token anzugeben, welches beim Starten des Docker-Containers angegeben wurde (geheimes-secret).
- Alle anderen Felder kann man leer lassen.

Nach einem Klick auf Speichern sollte eine Meldung erscheinen, dass die Einstellungen erfolgreich übernommen wurden.

Anschließend können Office-Dokumente ganz einfach direkt in der Cloud bearbeitet werden:

Online-Office mit Nextcloud und ONLYOFFICE

Update von ONLYOFFICE

Von Zeit zu Zeit erscheint ein neues Docker-Image für ONLYOFFICE. Leider ist es etwas umständlich herauszufinden, welche Version gerade installiert ist und ob es eine neue Container-Version gibt.

Um zu ermitteln, welche Version aktuell installiert ist, öffnet man am besten ein beliebiges Office-Dokument. Mit der Info-Schaltfläche wird die installierte Version von ONLYOFFICE angezeigt:

Über die Info-Schaltfläche kann die installierte Version ermittelt werden

Die Version des aktuellsten Images kann nun über Docker Hub (eine Art Suche für Docker-Images) ermittelt werden. Dazu sucht man einfach nach onlyoffice und lässt sich dann die Tags anzeigen. Schneller geht es über diesen Direktlink. Hier werden dann alle Versionen des Images angezeigt:

Die aktuelle Version im Docker Hub

Zwar werden in der ONLYOFFICE-Hilfe nur drei Versionsnummern angezeigt, allerdings sollte man auch anhand des Datums sehen können, wenn eine neue Version des Containers bereitsteht.

Das Update selbst kann man dann in wenigen Schritten durchführen:

Zunächst wird der aktuelle Container gestoppt und entfernt:

Shell

```
1 docker stop ONLYOFFICEDOCKER
2 docker rm ONLYOFFICEDOCKER
```

ONLYOFFICEDOCKER ist hier der Name des Containers, den wir beim ersten Start angegeben haben. Hier könnte man auch die ID des Containers angeben (kann man mit docker ps ermitteln).

Optional: Bei der Arbeit mit Docker bleiben oft Reste auf dem System zurück (z.B. alte, ungenutzte Container). Gerade wenn auf dem System ausschließlich ONLYOFFICE per Docker betrieben wird, empfiehlt es sich, das komplette Docker-System zu bereinigen. Dazu werden einfach folgende Befehle ausgeführt:

Shell

```
1 docker system prune -a
```

```
2 docker image prune -a

3 docker container prune
```

Anschließend wird die neuste Version des Containers heruntergeladen und über den bekannten Befehl gestartet:

Shell

```
1 docker pull onlyoffice/documentserver

2 docker run --name=ONLYOFFICEDOCKER
  -i -t -d -p 4433:443 -e
  JWT_ENABLED='true' -e
  JWT_SECRET='geheimes-secret'
```

```
--restart=always -v /app/
onlyoffice/
DocumentServer/data:/var/www/
onlyoffice/Data onlyoffice/
documentserver
```

Beim zweiten Befehl ist hier nur wichtig, dass genau das gleiche JWT_SECRET wie schon beim Start des Containers angegeben wird, ansonsten kann nach dem Update keine Verbindung mehr von Nextcloud hergestellt werden.

Troubleshooting

Manchmal kann es vorkommen, dass ONLYOFFICE nicht wie erwartet funktioniert. Bei dem System sind ziemlich viele Komponenten beteiligt (Webserver, Nextcloud, Docker-Container), so dass man zunächst einmal das Problem eingrenzen sollte.

Zunächst sollte immer erst die Subdomain für ONLYOFFICE aufgerufen werden: onlyoffice.meinedomain.de

Erscheint hier nicht die Übersichtsseite von ONLYOFFICE, dann tritt das Problem wohl auf dem Verbindungsweg vom Internet zum Webserver auf. Dann sollten zunächst einmal die Logs des Webservers kontrolliert werden (auf der Maschine, wo der Gateway-Host zu finden ist und auch Nextcloud installiert ist).

Shell

```
1                                   nano /var/log/nginx/error.log
```

Hier sollten dann Fehlermeldungen zu finden sein, mit den man das Problem weiter eingrenzen kann.

Wenn die Seite erreichbar ist, aber ONLYOFFICE trotzdem nicht funktioniert, sollte in diesem Fall zunächst einmal das Nextcloud-Log überprüft werden (in der Admin-Oberfläche von Nextcloud unter Protokollierung).

Falls hier auch nichts auffälliges zu sehen ist, sollten die Logs des Docker-Containers überprüft werden:

Shell

```
1                                   docker logs ONLYOFFICEDOCKER
```

Hier sind dann u.U. Hinweise zu finden, dass ein benötigter Dienst nicht gestartet werden konnte. In diesem Fall hilft es meistens, den Rechner, auf dem ONLYOFFICE läuft, einfach mal neu zu starten.

Erst wenn dies auch zu keinen neuen Erkenntnissen führt, kann man sich auch auf die Kommandozeile des Containers selbst einloggen. Dazu einfach auf der Docker-Maschine folgenden Befehl ausführen:

Shell

```
1 sudo docker exec -it ONLYOFFICEDOCKER bash
```

Auf der Kommandozeile des Containers kann dann eine detaillierte Fehlersuche stattfinden. Mit folgendem Befehl kann man beispielsweise die Webserver-Logs einzusehen:

Shell

```
1                                   nano /var/log/nginx/error.log
```

Um die Kommandozeile des Containers wieder zu verlassen, reicht folgender Befehl:

Shell

1 exit

Fazit

Es sind schon einige Schritte notwendig, um ONLYOFFICE in Verbindung mit Nextcloud zum Laufen zu kriegen. Dennoch lohnt es sich, da man nun direkt in der eigenen Cloud ein Online-Office nutzen kann. Wer vorher beispielsweise Google Docs verwendet hat, wird sich auch bei ONLYOFFICE sehr schnell zurechtfinden – und das ohne die neugierigen Blicke von Google & Co, da alle Daten sicher in der eigenen Nextcloud gespeichert sind.

Weiterführende Artikel

- Nextcloud auf Ubuntu Server 18.04 LTS mit nginx, MariaDB, PHP, Let’s Encrypt, Redis und Fail2ban
- Ubuntu Server 18.04 LTS als Hyper-V Gastsystem installieren und optimal einrichten
- Docker auf Ubuntu Server
- Nextcloud: Online-Office mit Collabora

Links

- Nextcloud-Homepage (englisch)
- Dynamisches DNS (Wikipedia)
- CNAME Resource-Record (Wikipedia)
- ONLYOFFICE-Homepage
- Docker-Homepage (englisch)
- Docker (Wikipedia)
- Get Docker-CE for Ubuntu (englisch)
- OpenDocument (Wikipedia)
- Office Open XML (Wikipedia)
- JSON Web Token (Wikipedia)

Source:
https://decatec.de/home-server/nextcloud-online-office-mit-onlyoffice-mit-eigener-subdomain/

Eindeutige ID: #1094
Verfasser: n/a
Letzte Änderung: 2019-11-17 11:46