Checkliste: ownCloud auf Ubuntu Server mit nginx, MariaDB und PHP

Dies ist eine Checkliste ohne weitere Erklärungen zur Installation von ownCloud auf Linux mit nginx, MariaDB und PHP.

Den kompletten Artikel mit weiteren Erklärungen zu den einzelnen Schritten zur Installation und Konfiguration gibt es unter [decatec.de]

Viele Aktionen setzen Root-Rechte voraus, daher geht die Liste davon aus, dass man (temporär) mit dauerhaften Root-Rechten arbeitet (*sudo -s*).

Hinweise zur Formatierung:

`Eingaben in der Kommandozeile`

```
Inhalte von Dateien
```

Hinweis, dass eine Anpassung der Eingaben notwendig ist

- Updates:
  - `apt-get update`
  - `apt-get upgrade`
  - `reboot`
- Statische IP-Adresse (hier ins Anpassungen an die eigene Netzwerkumgebung notwendig):
  - `nano /etc/network/interfaces`

```
auto eth0
iface eth0 inet static
address 192.168.178.20
netmask 255.255.255.0
network 192.168.178.0
broadcast 192.168.178.255
gateway 192.168.178.1
dns-nameservers 192.168.178.1
```

- nginx installieren:
  - `wget -O - http://nginx.org/keys/nginx_signing.key | apt-key add -`
  - `nano /etc/apt/sources.list`

```
# Nginx (Mainline)
deb http://nginx.org/packages/mainline/ubuntu/ vivid nginx
deb-src http://nginx.org/packages/mainline/ubuntu/ vivid nginx
```

  - `apt-get update`
  - `apt-get install nginx`
- MariaDB installieren:
  - `apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xcbcb082a1bb943db`
  - `nano /etc/apt/sources.list`

```
# MariaDB 10.0 repository list
# http://mariadb.org/mariadb/repositories/
deb [arch=amd64,i386]
http://ftp.hosteurope.de/mirror/mariadb.org/repo/10.0/ubuntu
wily main
```

```
deb-src
http://ftp.hosteurope.de/mirror/mariadb.org/repo/10.0/ubuntu
wily mainapt-get update
```

    - `apt-get install mariadb-server`
    - Root-Passwort vergeben
- PHP installieren:
    - `apt-get update`
    - `apt-get install php5-fpm php5-gd php5-json php5-mysql php5-curl php5-intl php5-mcrypt php5-imagick php5-apcu`
- SSL-Zertifikat generieren:
    - Es wird empfohlen, ein Zertifikat über Let's Encrypt zu erzeugen. Vorteil: Dieses wird von fast allen Browsern als vertrauenswürdig eingestuft und erzeugt keine Warnung (wie bei einem selbst signierten Zertifikat).
      Eine detaillierte Beschreibung zum Erzeugen von Zertifikaten über Let's Encrypt findet man unter decatec.de
    - Zugriffsrechte anpassen:
        - `chmod 600 /etc/letsencrypt/live/myserver.dnydns.org/fullchain.pem`
        - `chmod 600 /etc/letsencrypt/live/myserver.dnydns.org/privkey.pem`
        - `chmod 600 /etc/letsencrypt/live/myserver.dnydns.org/chain.pem`
        - `chmod 600 /etc/letsencrypt/live/myserver.dnydns.org/cert.pem`
    - Falls doch ein selbst signiertes Zertifikat zum Einsatz kommen soll:
        - `mkdir -p /etc/nginx/ssl`
        - `openssl req -new -x509 -days 365 -nodes -out /etc/nginx/ssl/certificate.crt -keyout /etc/nginx/ssl/certificate.key`
        - Angaben zum Zertifikat machen
        - `openssl dhparam -out /etc/nginx/ssl/dhparams.pem 2048`
        - Zugriffsrechte anpassen:
            - `chmod 600 /etc/nginx/ssl/certificate.crt`
            - `chmod 600 /etc/nginx/ssl/certificate.key`
            - `chmod 600 /etc/nginx/ssl/dhparams.pem`

- Konfiguration nginx:
    - Webroot + ownCloud Datenverzeichnis anlegen:
        - `mkdir -p /var/www`
        - `mkdir -p /var/oc_data`
        - `chown -R www-data:www-data /var/www`
        - `chown -R www-data:www-data /var/oc_data`
    - nginx-Konfiguration anpassen:
        - `nano /etc/nginx/nginx.conf`
            - `user  www-data www-data;`
            - `worker_processes auto;`
            - `server_tokens off;` (in der http-Konfiguration)
    - Default-Seite deaktivieren:

- `mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.disabled`
- `service nginx restart`

- Virtuellen Host für ownCloud anlegen:
  - `nano /etc/nginx/conf.d/myserver.dyndns.org.conf` (Anpassungen *server_name*/IP an die eigene Konfiguration notwendig)

```
upstream php-handler {
  server unix:/var/run/php5-fpm.sock;
  }

server {
  listen 80;
  server_name myserver.dyndns.org 192.168.178.20;
  # enforce https
  return 301 https://$server_name$request_uri;
  }

server {
  listen 443 ssl;
  server_name myserver.dyndns.org 192.168.178.20;

    ssl_certificate
/etc/letsencrypt/live/myserver.dyndns.org/fullchain.pem;
  ssl_certificate_key
/etc/letsencrypt/live/myserver.dyndns.org/privkey.pem;

  # Bei einem selbst signierten Zertifikat:
  #ssl_certificate /etc/nginx/ssl/certificate.crt;
  #ssl_certificate_key /etc/nginx/ssl/certificate.key;

  # Add headers to serve security related headers
  add_header Strict-Transport-Security "max-age=15768000;
includeSubDomains; preload;";
  add_header X-Content-Type-Options nosniff;
  add_header X-Frame-Options "SAMEORIGIN";
  add_header X-XSS-Protection "1; mode=block";
  add_header X-Robots-Tag none;

  ssl_dhparam /etc/nginx/ssl/dhparams.pem;
  ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-
GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-
SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-
SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-
AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-
SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-
AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-
AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-
SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-
SHA:AES:CAMELLIA:DES-CBC3-
SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-
DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA';
  ssl prefer server ciphers on;

  # Path to the root of your installation
  root /var/www/;

  index index.php index.html;

  # General PHP handler
```

```
  location ~ \.php$ {
   include fastcgi_params;
   fastcgi_param SCRIPT_FILENAME
$document root$fastcgi script name;
   fastcgi_param HTTPS on;
   fastcgi_pass php-handler;
   }

   #
   # ownCloud
   #
   rewrite ^/owncloud/caldav(.*)$ /owncloud/remote.php/caldav$1
redirect;
   rewrite ^/owncloud/carddav(.*)$
/owncloud/remote.php/carddav$1 redirect;
   rewrite ^/owncloud/webdav(.*)$ /owncloud/remote.php/webdav$1
redirect;

   location /owncloud {
    # set max upload size
    client_max_body_size 513M;

    fastcgi_buffers 64 4K;

    # Disable gzip to avoid the removal of the ETag header
    gzip off;

    # Uncomment if your server is build with the ngx_pagespeed
module This module is currently not supported.
    # pagespeed off;

    error_page 403 /owncloud/core/templates/403.php;
    error_page 404 /owncloud/core/templates/404.php;

    # The following 2 rules are only needed with webfinger
    rewrite ^/owncloud/.well-known/host-meta
/owncloud/public.php?service=host-meta last;
    rewrite ^/owncloud/.well-known/host-meta.json
/owncloud/public.php?service=host-meta-json last;
    rewrite ^/owncloud/.well-known/carddav
/owncloud/remote.php/carddav/ redirect;
    rewrite ^/owncloud/.well-known/caldav
/owncloud/remote.php/caldav/ redirect;

    rewrite ^(/owncloud/core/doc/[^\/]+/)$
/owncloud/$1/index.html;

    try_files $uri $uri/ /owncloud/index.php;

     # ownCloud specific PHP handler
     location ~ \.php(?:$|/) {
      fastcgi_split_path_info ^(.+\.php)(/.+)$;
      include fastcgi_params;
      fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
      fastcgi_param PATH_INFO $fastcgi_path_info;
      fastcgi_param HTTPS on;
      fastcgi_param modHeadersAvailable true; #Avoid sending
the security headers twice
      fastcgi_pass php-handler;
      fastcgi_read_timeout 300;
      fastcgi_param PHP_VALUE
"open_basedir=/var/www:/tmp/:/var/oc_data:/dev/urandom";
```

```
    }
  }

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

  location ~
^/owncloud/(?:\.htaccess|data|config|db_structure\.xml|README){
   deny all;
  }

  # Optional: set long EXPIRES header on static assets
  location ~*
^/owncloud(/.+\.(?:jpg|jpeg|gif|bmp|ico|png|css|js|swf))$ {
   expires 30d;
   # Optional: Don't log access to assets
   access_log off;
  }
}
```

  - `nginx -t`
  - `service nginx restart`
- Konfiguration MariaDB:
  - `mysql_secure_installation`
  - Alle Frage (bis auf Ändern des Root-Passwortes) mit Ja (*y*) beantworten
  - `nano /etc/mysql/my.cnf`
    - `binlog_format = MIXED` (unter [mysqld])
  - `service mysql restart`
- Konfiguration PHP:
  - `nano /etc/php5/fpm/php.ini`
    - `post_max_size = 513M`
    - `upload_max_filesize = 513M`
    - `cgi.fix_pathinfo = 0`
    - `open_basedir = /var/www/:/tmp/`
  - `nano /etc/php5/cli/php.ini`
    - `cgi.fix_pathinfo = 0`
    - `open_basedir = /var/www/:/tmp/:/var/oc_data`
    - `apc.enable_cli = 1`
  - `nano /etc/php5/fpm/pool.d/www.conf`
    - Kommentare der Umgebungsvariablen entfernen (nach *Pass environment variables like LD_LIBRARY_PATH. ALL &VARIABLES are taken from the current environment.* suchen)
  - `service php5-fpm restart`
- Download ownCloud + Verzeichnisrechte (jeweils aktuelle Version von ownCloud verwenden):
  - `wget https://download.owncloud.org/community/owncloud-8.2.1.tar.bz2`
  - `tar –xjf owncloud-8.2.1.tar.bz2 –C /var/www`
  - `rm owncloud-8.1.1.tar.bz2`
  - `chown -R www-data:www-data /var/www/owncloud`
  - `chown -R www-data:www-data /var/oc_data`
- Datenbank anlegen (Anpassungen bzgl. Datenbankname und User notwendig):
  - `mysql -u root -p`

    - `create database owncloud_db;`
    - `create user owncloud_user@localhost identified by 'MeInPasSw0rT';`
    - `grant all privileges on owncloud_db.* to owncloud_user@localhost;`
    - `flush privileges;`
    - `exit;`
- ownCloud Setup
    - Im Browser https://192.168.178.20/owncloud aufrufen (andere URL, wenn abweichende IP-Adresse verwendet wurde)
    - Zertifikat-Warnung kann ignoriert werden
    - Admin-Konto anlegen (mit den zuvor eingegebenen Daten: Datenverzeichnis, Datenbankname, Datenbank-Benutzer, Datenbank-Passwort)
- Installation Redis:
    - `apt-get install php5-redis redis-server`
    - `nano /etc/redis/redis.conf`
        - `unixsocket /var/run/redis/redis.sock`
        - `unixsocketperm 770`
    - Webserver-Benutzer zur Gruppe der Redis-Benutzer hinzufügen:
        - `usermod -a -G redis www-data`
- ownCloud Konfiguration anpassen:
    - `nano /var/www/owncloud/config/config.php` (*instanceid*, *passwordsalt* und *secret* ausgelassen; Anpassungen an eigen IP/DynDNS-Adresse notwendig)

```
<?php
$CONFIG = array (
  'instanceid' => '...',
  'passwordsalt' => '...',
  'secret' => '...',
  'trusted_domains' =>
  array (
    0 => '192.168.178.20',
    1 => 'myserver.dyndns.org',
  ),
  'datadirectory' => '/var/oc_data',
  'overwrite.cli.url' => 'https://192.168.178.20/owncloud',
  'dbtype' => 'mysql',
  'version' => '8.2.1.4',
  'dbname' => 'owncloud_db',
  'dbhost' => 'localhost',
  'dbtableprefix' => 'oc_',
  'dbuser' => 'owncloud_user',
  'dbpassword' => 'MeInPasSw0rT',
  'logtimezone' => 'Europe/Berlin',
  'installed' => true,
  'memcache.local' => '\OC\Memcache\APCu',
  'filelocking.enabled' => 'true',
  'memcache.locking' => '\OC\Memcache\Redis',
  'redis' => array(
    'host' => '/var/run/redis/redis.sock',
    'port' => 0,
    'timeout' => 0.0,
   ),
);
```

- Zugriffsrechte anpassen:

- `find /var/www/owncloud/ -type f -print0 | xargs -0 chmod 0640`
- `find /var/www/owncloud/ -type d -print0 | xargs -0 chmod 0750`
- `chown -R root:www-data /var/www/owncloud/`
- `chown -R www-data:www-data /var/www/owncloud/apps/`
- `chown -R www-data:www-data /var/www/owncloud/config/`
- `chown -R www-data:www-data /var/www/owncloud/themes/`
- `chown -R www-data:www-data /var/oc_data/`
- `chown root:www-data /var/www/owncloud/.htaccess`
- `chown root:www-data /var/www/owncloud/data/.htaccess`
- `chmod 0644 /var/www/owncloud/.htaccess`
- `chmod 0644 /var/oc_data/.htaccess`

- Cronjob einrichten:
  - crontab -u www-data –e

```
*/15  *  *  *  * php -f /var/www/owncloud/cron.php > /dev/null 2>&1
```

- Abschlussarbeiten (siehe Artikel):
  - Port-Forwarding einrichten
  - Zertifikat installieren