

Nextcloud

Nextcloud ist eine Reihe von Client-Server-Software zum Erstellen und Verwenden von Dateihosting-Diensten. Es ähnelt funktional der Dropbox, obwohl Nextcloud kostenlos und open-source ist und es jedem ermöglicht, es auf einem privaten Server zu installieren und zu bedienen. Im Gegensatz zu proprietären Diensten wie Dropbox ermöglicht die offene Architektur dem Server zusätzliche Funktionen in Form von Anwendungen.

[Maintenance und Release Schedule](#)

Pakete

```
pacman -S nextcloud php php-fpm php-gd php-imagick php-apcu php-igbinary  
php-redis ffmpeg libreoffice
```

Redis

Installiere [redis](#).

Ordner erstellen

Datenspeicher

Die Speicherung aller deiner Daten wird auf einem anderen Laufwerk platziert.

```
mkdir -p /mnt/wolfs/nextcloud/data  
chown -R nextcloud: /mnt/wolfs/nextcloud/  
chmod 770 /mnt/wolfs/nextcloud/data
```

Vergesse nicht, den Speicherpfad im folgenden Tutorial zu ändern, wenn du ihn ändern solltest.

Session

```
install --owner=nextcloud --group=nextcloud --mode=700 -d  
/var/lib/nextcloud/sessions
```

PHP

php.ini für occ commands

```
cp /etc/php/php.ini /etc/webapps/nextcloud/php.ini
chown nextcloud:nextcloud /etc/webapps/nextcloud/php.ini
nano /etc/webapps/nextcloud/php.ini
```

```
date.timezone = Pacific/Auckland
memory_limit = 2048M
open_basedir =
/mnt/wolfs/nextcloud:/var/lib/nextcloud:/tmp:/usr/share/webapps/nextcloud:/e
tc/webapps/nextcloud:/dev/urandom:/usr/lib/php/modules:/var/log/nextcloud:/p
roc/meminfo:/run/redis
extension=bcmath
extension=bz2
extension=exif
extension=gd
extension=iconv
extension=intl
extension=pdo_mysql
extension=imagick
extension=sysvsem
;extension=igbinary
;extension=redis
extension=apcu
apc.ttl=7200
apc.enable_cli = 1
```

```
export NEXTCLOUD_PHP_CONFIG=/etc/webapps/nextcloud/php.ini
nano ~/.bashrc
```

```
export NEXTCLOUD_PHP_CONFIG=/etc/webapps/nextcloud/php.ini
```

Aktiviere opcache

```
cp /etc/php/php.ini /etc/php/php-fpm.ini
chmod o+r /etc/php/php-fpm.ini
nano /etc/php/php-fpm.ini
```

```
zend_extension=opcache

[opcache]
opcache.enable = 1
opcache.interned_strings_buffer = 32
opcache.max_accelerated_files = 10000
opcache.memory_consumption = 1024
```

```
opcache.save_comments = 1
opcache.revalidate_freq = 1
```

nextcloud.conf pool Datei

```
nano /etc/php/php-fpm.d/nextcloud.conf
```

```
; Start a new pool named 'nextcloud'.
; the variable $pool can be used in any directive and will be replaced by
the
; pool name ('nextcloud' here)
[nextcloud]

; Per pool prefix
; It only applies on the following directives:
; - 'access.log'
; - 'slowlog'
; - 'listen' (unixsocket)
; - 'chroot'
; - 'chdir'
; - 'php_values'
; - 'php_admin_values'
; When not set, the global prefix (or /usr) applies instead.
; Note: This directive can also be relative to the global prefix.
; Default Value: none
;prefix = /path/to/pools/$pool

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default user's
group
;       will be used.
user = nextcloud
group = nextcloud

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'    - to listen on a TCP socket to a specific IPv4
address on
;                           a specific port;
;   '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6
address on
;                           a specific port;
;   'port'                 - to listen on a TCP socket to all addresses
;                           (IPv6 and IPv4-mapped) on a specific port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /run/php-fpm/nextcloud.sock

; Set listen(2) backlog.
; Default Value: 511 (-1 on FreeBSD and OpenBSD)
```

```
;listen.backlog = 511

; Set permissions for unix socket, if one is used. In Linux, read/write
; permissions must be set in order to allow connections from a web server.
Many
; BSD-derived systems allow connections regardless of permissions. The owner
; and group can be specified either by name or by their numeric IDs.
; Default Values: user and group are set as the running user
;
; mode is set to 0660
listen.owner = nextcloud
listen.group = http
listen.mode = 0660
; When POSIX Access Control Lists are supported you can set them using
; these options, value is a comma separated list of user/group names.
; When set, listen.owner and listen.group are ignored
;listen.acl_users =
;listen.acl_groups =

; List of addresses (IPv4/IPv6) of FastCGI clients which are allowed to
connect.
; Equivalent to the FCGI_WEB_SERVER_ADDRS environment variable in the
original
; PHP FCGI (5.2.2+). Makes sense only with a tcp listening socket. Each
address
; must be separated by a comma. If this value is left blank, connections
will be
; accepted from any ip address.
; Default Value: any
;listen.allowed_clients = 127.0.0.1

; Specify the nice(2) priority to apply to the pool processes (only if set)
; The value can vary from -19 (highest priority) to 20 (lower priority)
; Note: - It will only work if the FPM master process is launched as root
;       - The pool processes will inherit the master process priority
;       unless it specified otherwise
; Default Value: no set
; process.priority = -19

; Set the process dumpable flag (PR_SET_DUMPABLE prctl) even if the process
user
; or group is different than the master process user. It allows to create
process
; core dump and ptrace the process for the pool user.
; Default Value: no
; process.dumpable = yes

; Choose how the process manager will control the number of child processes.
; Possible Values:
;   static - a fixed number (pm.max_children) of child processes;
;   dynamic - the number of child processes are set dynamically based on the
;             following directives. With this process management, there will
```

```

be
;          always at least 1 children.
;          pm.max_children      - the maximum number of children that can
;                                be alive at the same time.
;          pm.start_servers     - the number of children created on
startup.
;          pm.min_spare_servers - the minimum number of children in
'idle'
;                                state (waiting to process). If the
number
;                                of 'idle' processes is less than this
;                                number then some children will be
created.
;          pm.max_spare_servers - the maximum number of children in
'idle'
;                                state (waiting to process). If the
number
;                                of 'idle' processes is greater than
this
;                                number then some children will be
killed.
;          pm.max_spawn_rate    - the maximum number of rate to spawn
child
;                                processes at once.
;  ondemand - no children are created at startup. Children will be forked
when
;          new requests will connect. The following parameter are used:
;          pm.max_children      - the maximum number of children
that
;                                can be alive at the same time.
;          pm.process_idle_timeout - The number of seconds after which
;                                an idle process will be killed.
; Note: This value is mandatory.
pm = dynamic

; The number of child processes to be created when pm is set to 'static' and
the
; maximum number of child processes when pm is set to 'dynamic' or
'ondemand'.
; This value sets the limit on the number of simultaneous requests that will
be
; served. Equivalent to the ApacheMaxClients directive with mpm_prefork.
; Equivalent to the PHP_FCGI_CHILDREN environment variable in the original
PHP
; CGI. The below defaults are based on a server without much resources.
Don't
; forget to tweak pm.* to fit your needs.
; Note: Used when pm is set to 'static', 'dynamic' or 'ondemand'
; Note: This value is mandatory.
pm.max_children = 15

```

```
; The number of child processes created on startup.
; Note: Used only when pm is set to 'dynamic'
; Default Value: (min_spare_servers + max_spare_servers) / 2
pm.start_servers = 2

; The desired minimum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.min_spare_servers = 1

; The desired maximum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.max_spare_servers = 3

; The number of rate to spawn child processes at once.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
; Default Value: 32
pm.max_spawn_rate = 32

; The number of seconds after which an idle process will be killed.
; Note: Used only when pm is set to 'ondemand'
; Default Value: 10s
pm.process_idle_timeout = 10s;

; The number of requests each child process should execute before
respawning.
; This can be useful to work around memory leaks in 3rd party libraries. For
; endless request processing specify '0'. Equivalent to
PHP_FCGI_MAX_REQUESTS.
; Default Value: 0
pm.max_requests = 500

; The URI to view the FPM status page. If this value is not set, no URI will
be
; recognized as a status page. It shows the following information:
;   pool                - the name of the pool;
;   process manager      - static, dynamic or ondemand;
;   start time           - the date and time FPM has started;
;   start since          - number of seconds since FPM has started;
;   accepted conn        - the number of request accepted by the pool;
;   listen queue         - the number of request in the queue of pending
;                           connections (see backlog in listen(2));
;   max listen queue     - the maximum number of requests in the queue
;                           of pending connections since FPM has started;
;   listen queue len     - the size of the socket queue of pending
connections;
;   idle processes       - the number of idle processes;
;   active processes     - the number of active processes;
;   total processes      - the number of idle + active processes;
```

```
; max active processes - the maximum number of active processes since FPM
;                          has started;
; max children reached - number of times, the process limit has been
reached,
;                          when pm tries to start more children (works only
for
;                          pm 'dynamic' and 'ondemand');
; Value are updated in real time.
; Example output:
; pool:                  www
; process manager:       static
; start time:            01/Jul/2011:17:53:49 +0200
; start since:           62636
; accepted conn:         190460
; listen queue:          0
; max listen queue:      1
; listen queue len:      42
; idle processes:        4
; active processes:      11
; total processes:       15
; max active processes:  12
; max children reached:  0
;
; By default the status page output is formatted as text/plain. Passing
either
; 'html', 'xml' or 'json' in the query string will return the corresponding
; output syntax. Example:
; http://www.foo.bar/status
; http://www.foo.bar/status?json
; http://www.foo.bar/status?html
; http://www.foo.bar/status?xml
;
; By default the status page only outputs short status. Passing 'full' in
the
; query string will also return status for each pool process.
; Example:
; http://www.foo.bar/status?full
; http://www.foo.bar/status?json&full
; http://www.foo.bar/status?html&full
; http://www.foo.bar/status?xml&full
; The Full status returns for each process:
; pid                    - the PID of the process;
; state                  - the state of the process (Idle, Running, ...);
; start time             - the date and time the process has started;
; start since            - the number of seconds since the process has
started;
; requests               - the number of requests the process has served;
; request duration       - the duration in µs of the requests;
; request method         - the request method (GET, POST, ...);
; request URI            - the request URI with the query string;
; content length         - the content length of the request (only with
```

```
POST);
; user - the user (PHP_AUTH_USER) (or '-' if not set);
; script - the main script called (or '-' if not set);
; last request cpu - the %cpu the last request consumed
; it's always 0 if the process is not in Idle state
; because CPU calculation is done when the request
; processing has terminated;
; last request memory - the max amount of memory the last request
consumed
; it's always 0 if the process is not in Idle state
; because memory calculation is done when the
request
; processing has terminated;
; If the process is in Idle state, then informations are related to the
; last request the process has served. Otherwise informations are related to
; the current request being served.
; Example output:
; *****
; pid: 31330
; state: Running
; start time: 01/Jul/2011:17:53:49 +0200
; start since: 63087
; requests: 12808
; request duration: 1250261
; request method: GET
; request URI: /test_mem.php?N=10000
; content length: 0
; user: -
; script: /home/fat/web/docs/php/test_mem.php
; last request cpu: 0.00
; last request memory: 0
;
; Note: There is a real-time FPM status monitoring sample web page available
; It's available in: /usr/share/php/fpm/status.html
;
; Note: The value must start with a leading slash (/). The value can be
; anything, but it may not be a good idea to use the .php extension or
it
; may conflict with a real PHP file.
; Default Value: not set
;pm.status_path = /status

; The address on which to accept FastCGI status request. This creates a new
; invisible pool that can handle requests independently. This is useful
; if the main pool is busy with long running requests because it is still
possible
; to get the status before finishing the long running requests.
;
; Valid syntaxes are:
; 'ip.add.re.ss:port' - to listen on a TCP socket to a specific IPv4
address on
```



```
;                                a specific port;
; '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6
address on
;                                a specific port;
; 'port'                        - to listen on a TCP socket to all addresses
;                                (IPv6 and IPv4-mapped) on a specific port;
; '/path/to/unix/socket' - to listen on a unix socket.
; Default Value: value of the listen option
;pm.status_listen = 127.0.0.1:9001

; The ping URI to call the monitoring page of FPM. If this value is not set,
no
; URI will be recognized as a ping page. This could be used to test from
outside
; that FPM is alive and responding, or to
; - create a graph of FPM availability (rrd or such);
; - remove a server from a group if it is not responding (load balancing);
; - trigger alerts for the operating team (24/7).
; Note: The value must start with a leading slash (/). The value can be
;       anything, but it may not be a good idea to use the .php extension or
it
;       may conflict with a real PHP file.
; Default Value: not set
;ping.path = /ping

; This directive may be used to customize the response of a ping request.
The
; response is formatted as text/plain with a 200 response code.
; Default Value: pong
;ping.response = pong

; The access log file
; Default: not set
;access.log = log/$pool.access.log
access.log = /var/log/nextcloud/$pool.log

; The access log format.
; The following syntax is allowed
; %: the '%' character
; %C: %CPU used by the request
;     it can accept the following format:
;     - %{user}C for user CPU only
;     - %{system}C for system CPU only
;     - %{total}C  for user + system CPU (default)
; %d: time taken to serve the request
;     it can accept the following format:
;     - %{seconds}d (default)
;     - %{milliseconds}d
;     - %{milli}d
;     - %{microseconds}d
;     - %{micro}d
```

```

; %e: an environment variable (same as $_ENV or $_SERVER)
;     it must be associated with embraces to specify the name of the env
;     variable. Some examples:
;     - server specifics like: %{REQUEST_METHOD}e or %{SERVER_PROTOCOL}e
;     - HTTP headers like: %{HTTP_HOST}e or %{HTTP_USER_AGENT}e
; %f: script filename
; %l: content-length of the request (for POST request only)
; %m: request method
; %M: peak of memory allocated by PHP
;     it can accept the following format:
;     - %{bytes}M (default)
;     - %{kilobytes}M
;     - %{kilo}M
;     - %{megabytes}M
;     - %{mega}M
; %n: pool name
; %o: output header
;     it must be associated with embraces to specify the name of the
header:
;     - %{Content-Type}o
;     - %{X-Powered-By}o
;     - %{Transfert-Encoding}o
;     - ....
; %p: PID of the child that serviced the request
; %P: PID of the parent of the child that serviced the request
; %q: the query string
; %Q: the '?' character if query string exists
; %r: the request URI (without the query string, see %q and %Q)
; %R: remote IP address
; %s: status (response code)
; %t: server time the request was received
;     it can accept a strftime(3) format:
;     %d/%b/%Y:%H:%M:%S %z (default)
;     The strftime(3) format must be encapsulated in a
%{<strftime_format>}t tag
;     e.g. for a ISO8601 formatted timestring, use: %{Y-%m-%dT%H:%M:%S%z}t
; %T: time the log has been written (the request has finished)
;     it can accept a strftime(3) format:
;     %d/%b/%Y:%H:%M:%S %z (default)
;     The strftime(3) format must be encapsulated in a
%{<strftime_format>}t tag
;     e.g. for a ISO8601 formatted timestring, use: %{Y-%m-%dT%H:%M:%S%z}t
; %u: remote user
;
; Default: "%R - %u %t \"%m %r\" %s"
;access.format = "%R - %u %t \"%m %r%Q%q\" %s %f %{milli}d %{kilo}M %C%"
access.format = "%{Y-%m-%dT%H:%M:%S%z}t %R: \"%m %r%Q%q\" %s %f %{milli}d
%{kilo}M %C%"

; The log file for slow requests
; Default Value: not set

```

```
; Note: slowlog is mandatory if request_slowlog_timeout is set
;slowlog = log/$pool.log.slow

; The timeout for serving a single request after which a PHP backtrace will
be
; dumped to the 'slowlog' file. A value of '0s' means 'off'.
; Available units: s(econds)(default), m(inutes), h(ours), or d(ays)
; Default Value: 0
;request_slowlog_timeout = 0

; Depth of slow log stack trace.
; Default Value: 20
;request_slowlog_trace_depth = 20

; The timeout for serving a single request after which the worker process
will
; be killed. This option should be used when the 'max_execution_time' ini
option
; does not stop script execution for some reason. A value of '0' means
'off'.
; Available units: s(econds)(default), m(inutes), h(ours), or d(ays)
; Default Value: 0
;request_terminate_timeout = 0

; The timeout set by 'request_terminate_timeout' ini option is not engaged
after
; application calls 'fastcgi_finish_request' or when application has
finished and
; shutdown functions are being called (registered via
register_shutdown_function).
; This option will enable timeout limit to be applied unconditionally
; even in such cases.
; Default Value: no
;request_terminate_timeout_track_finished = no

; Set open file descriptor rlimit.
; Default Value: system defined value
;rlimit_files = 1024

; Set max core size rlimit.
; Possible Values: 'unlimited' or an integer greater or equal to 0
; Default Value: system defined value
;rlimit_core = 0

; Chroot to this directory at the start. This value must be defined as an
; absolute path. When this value is not set, chroot is not used.
; Note: you can prefix with '$prefix' to chroot to the pool prefix or one
; of its subdirectories. If the pool prefix is not set, the global prefix
; will be used instead.
; Note: chrooting is a great security feature and should be used whenever
; possible. However, all PHP paths will be relative to the chroot
```

```
; (error_log, sessions.save_path, ...).
; Default Value: not set
;chroot =

; Chdir to this directory at the start.
; Note: relative path can be used.
; Default Value: current directory or / when chroot
;chdir = /srv/http
chdir = /usr/share/webapps/$pool

; Redirect worker stdout and stderr into main error log. If not set, stdout
and
; stderr will be redirected to /dev/null according to FastCGI specs.
; Note: on highloaded environment, this can cause some delay in the page
; process time (several ms).
; Default Value: no
;catch_workers_output = yes

; Decorate worker output with prefix and suffix containing information about
; the child that writes to the log and if stdout or stderr is used as well
as
; log level and time. This options is used only if catch_workers_output is
yes.
; Settings to "no" will output data as written to the stdout or stderr.
; Default value: yes
;decorate_workers_output = no

; Clear environment in FPM workers
; Prevents arbitrary environment variables from reaching FPM worker
processes
; by clearing the environment in workers before env vars specified in this
; pool configuration are added.
; Setting to "no" will make all environment variables available to PHP code
; via getenv(), $_ENV and $_SERVER.
; Default Value: yes
;clear_env = no

; Limits the extensions of the main script FPM will allow to parse. This can
; prevent configuration mistakes on the web server side. You should only
limit
; FPM to .php extensions to prevent malicious users to use other extensions
to
; execute php code.
; Note: set an empty value to allow all extensions.
; Default Value: .php
;security.limit_extensions = .php .php3 .php4 .php5 .php7

; Pass environment variables like LD_LIBRARY_PATH. All $VARIABLEs are taken
from
; the current environment.
; Default Value: clean env
```

```
env[HOSTNAME] = $HOSTNAME
env[PATH] = /usr/local/bin:/usr/bin
env[TMP] = /tmp
env[TMPDIR] = /tmp
env[TEMP] = /tmp

; Additional php.ini defines, specific to this pool of workers. These
settings
; overwrite the values previously defined in the php.ini. The directives are
the
; same as the PHP SAPI:
;   php_value/php_flag          - you can set classic ini defines which
can
;                               be overwritten from PHP call 'ini_set'.
;   php_admin_value/php_admin_flag - these directives won't be overwritten
by
;                               PHP call 'ini_set'
; For php_*flag, valid values are on, off, 1, 0, true, false, yes or no.

; Defining 'extension' will load the corresponding shared extension from
; extension_dir. Defining 'disable_functions' or 'disable_classes' will not
; overwrite previously defined php.ini values, but will append the new value
; instead.

; Note: path INI options can be relative and will be expanded with the
prefix
; (pool, global or /usr)

; Default Value: nothing is defined by default except the values in php.ini
and
;               specified at startup with the -d argument
;php_admin_value[sendmail_path] = /usr/sbin/sendmail -t -i -f
www@my.domain.com
;php_flag[display_errors] = off
;php_admin_value[error_log] = /var/log/fpm-php.www.log
;php_admin_flag[log_errors] = on
;php_admin_value[memory_limit] = 32M

php_value[date.timezone] = Pacific/Auckland

php_value[open_basedir] =
/mnt/wolfs/nextcloud:/var/lib/nextcloud:/tmp:/usr/share/webapps/nextcloud:/e
tc/webapps/nextcloud:/dev/urandom:/usr/lib/php/modules:/var/log/nextcloud:/p
roc/meminfo:/run/redis

; put session data in dedicated directory
php_value[session.save_path] = /var/lib/$pool/sessions
php_value[session.gc_maxlifetime] = 21600
php_value[session.gc_divisor] = 500
php_value[session.gc_probability] = 1
```

```
php_flag[expose_php] = false
php_value[post_max_size] = 1000G
php_value[upload_max_filesize] = 1000G

; as recommended in admin manual (avoids related warning in admin GUI later)
php_flag[output_buffering] = off
php_value[max_input_time] = 3600
php_value[max_execution_time] = 3600

php_value[memory_limit] = 2048M

; opcache settings must be defined in php-fpm.ini. otherwise (i.e. when
defined here)
; this causes segmentation faults in php-fpm worker processes

; uncomment if php-apcu is installed and used
php_value[extension] = apcu
; (see https://github.com/krakjoe/apcu/blob/simplify/INSTALL)
php_value[apc.ttl] = 7200
php_flag[apc.enable_cli] = 1

php_value[extension] = bcmath
php_value[extension] = bz2
php_value[extension] = exif
php_value[extension] = gd
php_value[extension] = gmp
; uncomment if php-imagick is installed and used
php_value[extension] = imagick
; recommended to enable
php_value[extension] = intl
php_value[extension] = iconv
; uncomment if php-memcached is installed and used
; php_value[extension] = memcached
; uncomment exactly one of the pdo extensions
php_value[extension] = pdo_mysql
; php_value[extension] = pdo_pgsql
; php_value[extension] = pdo_sqlite
; uncomment if php-igbinary is installed and used
; php_value[extension] = igbinary
; uncomment if php-redis is installed and used (requires php-igbinary)
; php_value[extension] = redis
; uncomment if php-xsl is installed and used
; php_value[extension] = xsl
php_value[extension] = sysvsem
```

```
chmod o+r /etc/php/php-fpm.d/nextcloud.conf
```

Überschreibe php-fpm service

```
systemctl edit php-fpm.service
```

```
[Service]
ExecStart=
ExecStart=/usr/bin/php-fpm --nodaemonize --fpm-config /etc/php/php-fpm.conf
--php-ini /etc/php/php-fpm.ini
ReadWritePaths=/var/lib/nextcloud
ReadWritePaths=/etc/webapps/nextcloud/config
ReadWritePaths=/mnt/wolfs/nextcloud
```

MariaDB

Check auch [mariadb](#)

Datenbank und Benutzer erstellen

```
mysql -u root -p
```

```
CREATE USER 'nextcloud'@'localhost' IDENTIFIED BY 'db-password';
CREATE DATABASE IF NOT EXISTS nextcloud CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
GRANT ALL PRIVILEGES on nextcloud.* to 'nextcloud'@'localhost';
FLUSH privileges;
```

mysql.conf

```
nano /etc/my.cnf.d/server.cnf
```

```
[mysqld]
skip_networking
transaction_isolation=READ-COMMITTED
```

Setup Nextcloud's database schema

Änder alles, was für dich geändert werden muss.

```
occ maintenance:install \
  --database=mysql \
  --database-name=nextcloud \
  --database-host=localhost:/run/mysqld/mysqld.sock \
  --database-user=nextcloud \
  --database-pass=db-password \
  --admin-pass=admin-password \
  --admin-email=admin-email \
```

```
--data-dir=/mnt/wolfs/nextcloud/
```

SSL

Check [ssl](#)

Nginx

Check auch [nginx](#).

```
nano /etc/nginx/sites-available/nextcloud.home
```

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

# Set the `immutable` cache control options only for assets with a cache
busting `v` argument
map $arg_v $asset_immutable {
    "" "";
    default "immutable";
}

server {
    listen 80;
#    listen [::]:80;
    server_name nextcloud.home;

    # enforce https
    return 301 https://$server_name:443$request_uri;
}

server {
    listen 443 ssl http2;
#    listen [::]:443 ssl http2;
    server_name nextcloud.home;

    ssl_certificate /etc/nginx/ssl/nextcloud.home.crt;
    ssl_certificate_key /etc/nginx/ssl/nextcloud.home.key;
    include conf.d/ssl-params.conf

    # Path to the root of your installation
    root /usr/share/webapps/nextcloud;

    access_log /var/log/nginx/nextcloud.access.log;
    error_log /var/log/nginx/nextcloud.error.log;
```



```
# HSTS settings
# WARNING: Only add the preload option once you read about
# the consequences in https://hstspreload.org/. This option
# will add the domain to a hardcoded list that is shipped
# in all major browsers and getting removed from this list
# could take several months.
#add_header Strict-Transport-Security "max-age=15768000;
includeSubDomains; preload" always;

# set max upload size and increase upload timeout:
client_max_body_size 512M;
client_body_timeout 300s;
fastcgi_buffers 64 4K;

# Enable gzip but do not remove ETag headers
gzip on;
gzip_vary on;
gzip_comp_level 4;
gzip_min_length 256;
gzip_proxied expired no-cache no-store private no_last_modified
no_etag auth;
gzip_types application/atom+xml application/javascript
application/json application/ld+json application/manifest+json
application/rss+xml application/vnd.geo+json application/vnd.ms-fontobject
application/wasm application/x-font-ttf application/x-web-app-manifest+json
application/xhtml+xml application/xml font/opentype image/bmp image/svg+xml
image/x-icon text/cache-manifest text/css text/plain text/vcard
text/vnd.rim.location.xloc text/vtt text/x-component text/x-cross-domain-
policy;

# Pagespeed is not supported by Nextcloud, so if your server is
built
# with the `ngx_pagespeed` module, uncomment this line to disable
it.
#pagespeed off;

# The settings allows you to optimize the HTTP2 bandwidth.
# See
https://blog.cloudflare.com/delivering-http-2-upload-speed-improvements/
# for tuning hints
client_body_buffer_size 512k;

# HTTP response headers borrowed from Nextcloud `.htaccess`
add_header Referrer-Policy "no-referrer"
always;
add_header X-Content-Type-Options "nosniff"
always;
add_header X-Download-Options "noopen"
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;

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

    # Add .mjs as a file extension for javascript
    # Either include it in the default mime.types list
    # or include you can include that list explicitly and add the file
extension
    # only for Nextcloud like below:
    include mime.types;
    types {
        application/javascript js mjs;
    }

    # Specify how to handle directories -- specifying
`/index.php$request_uri`
    # here as the fallback means that Nginx always exhibits the desired
behaviour
    # when a client requests a path that corresponds to a directory that
exists
    # on the server. In particular, if that directory contains an
index.php file,
    # that file is correctly served; if it doesn't, then the request is
passed to
    # the front-end controller. This consistent behaviour means that we
don't need
    # to specify custom rules for certain paths (e.g. images and other
assets,
    # `/updater`, `/ocm-provider`, `/ocs-provider`), and thus
    # `try_files $uri $uri/ /index.php$request_uri`
    # always provides the desired behaviour.
    index index.php index.html /index.php$request_uri;

    # Rule borrowed from `.htaccess` to handle Microsoft DAV clients
    location = / {
        if ( $http_user_agent ~ ^DavClnt ) {
            return 302 /remote.php/webdav/$is_args$args;
        }
    }

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

```
# Make a regex exception for `/.well-known` so that clients can
still
# access it despite the existence of the regex rule
# `location ~ /\.(|autotest|...)` which would otherwise handle
requests
# for `/.well-known`.
location ^~ /.well-known {
    # The rules in this block are an adaptation of the rules
    # in `.htaccess` that concern `/.well-known`.

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

    location /.well-known/acme-challenge    { try_files $uri $uri/
=404; }
    location /.well-known/pki-validation    { try_files $uri $uri/
=404; }

    # Let Nextcloud's API for `/.well-known` URIs handle all other
    # requests by passing them to the front-end controller.
    return 301 /index.php$request_uri;
}

# Rules borrowed from `.htaccess` to hide certain paths from clients
location ~
^/(?::build|tests|config|lib|3rdparty|templates|data)(?::$|/) { return 404; }
    location ~ ^/(?::\.|autotest|occ|issue|indie|db_|console)
{ return 404; }

# Ensure this block, which passes PHP files to the PHP process, is
above the blocks
# which handle static assets (as seen below). If this block is not
declared first,
# then Nginx will encounter an infinite rewriting loop when it
prepends `/index.php`
# to the URI, resulting in a HTTP 500 error response.
location ~ \.php(?:$|/) {
    # Required for legacy support
    rewrite
^/(?!(index|remote|public|cron|core/ajax/update|status|ocs/v[12]|updater\|
.+|oc[ms]-provider\|.+\|.+\/richdocumentscode\/proxy) /index.php$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;          # Avoid sending
the security headers twice
        fastcgi_param front_controller_active true;      # Enable pretty
urls
        fastcgi_pass php-handler;

        fastcgi_intercept_errors on;
        fastcgi_request_buffering off;

        fastcgi_max_temp_file_size 0;
    }

    location ~ \.(?:css|js|svg|gif|png|jpg|ico|wasm|tflite|map)$ {
        try_files $uri /index.php$request_uri;
        add_header Cache-Control "public, max-age=15778463,
$asset_immutable";
        access_log off;          # Optional: Don't log access to assets

        location ~ \.wasm$ {
            default_type application/wasm;
        }
    }

    location ~ \.woff2?$ {
        try_files $uri /index.php$request_uri;
        expires 7d;              # Cache-Control policy borrowed from
`.htaccess`
        access_log off;          # Optional: Don't log access to assets
    }

    # Rule borrowed from `.htaccess`
    location /remote {
        return 301 /remote.php$request_uri;
    }

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

```
ln -s /etc/nginx/sites-available/nextcloud.home /etc/nginx/sites-enabled/
systemctl reload nginx.service
```

Hintergrundjobs

```
systemctl edit nextcloud-cron.service
```

```
[Service]
ExecStart=
ExecStart=/usr/bin/php -c /etc/webapps/nextcloud/php.ini -f
/usr/share/webapps/nextcloud/cron.php
```

```
systemctl enable --now nextcloud-cron.timer
```

Nextcloud config

```
nano /etc/webapps/nextcloud/config/config.php
```

Vertrauenswürdige Domänen

```
'trusted_domains' =>
array (
    0 => 'localhost',
    1 => 'nextcloud.home',
),
'overwrite.cli.url' => 'https://nextcloud.home',
'htaccess.RewriteBase' => '/',
```

Nur APCu

```
'memcache.local' => '\OC\Memcache\APCu',
```

Redis & APCu

Mit socket

Im Moment musst du die Berechtigung für die Socket-Datei in `/etc/redis/redis.conf` auf `UnixsocketPerm 777` ändern.

```
'memcache.local' => '\OC\Memcache\APCu',
'memcache.distributed' => '\OC\Memcache\Redis',
'memcache.locking' => '\OC\Memcache\Redis',
'filelocking.enabled' => 'true',
'redis' =>
array (
    'host'      => '/run/redis/redis.sock',
    'port'      => 0,
```

```
'dbindex'    => 0,  
'password'   => 'your-password',  
'timeout'    => 1.5,  
) ,
```

Ohne socket

```
'memcache.local' => '\0C\Memcache\APCu',  
'memcache.distributed' => '\0C\Memcache\Redis',  
'memcache.locking' => '\0C\Memcache\Redis',  
'filelocking.enabled' => 'true',  
'redis' =>  
array (  
    'host' => 'localhost',  
    'port'   => 6379,  
    'password' => 'your-password',  
) ,
```

Update via pacman hook

```
mkdir -vp /etc/pacman.d/hooks  
cp -a /usr/share/doc/nextcloud/nextcloud.hook /etc/pacman.d/hooks/10-  
nextcloud.hook  
nano /etc/pacman.d/hooks/10-nextcloud.hook
```

```
# Update Nextcloud when core or -apps are touched
```

```
[Trigger]  
Operation = Install  
Operation = Upgrade  
Type = Package  
Target = nextcloud  
Target = nextcloud-app-*  
  
[Action]  
Description = Updating Nextcloud installation  
When = PostTransaction  
Exec = /usr/bin/runuser -u nextcloud -- /usr/bin/php --php-ini  
/etc/webapps/nextcloud/php.ini /usr/share/webapps/nextcloud/occ upgrade
```

Collabora

Check [docker](#)

```
docker run -t -d -p 127.0.0.1:9980:9980 -e  
"aliasgroup1=https://nextcloud.home:443" -e "username=your-user" -e
```

```
"password=your-password" --restart always collabora/code
```

Nginx

Check auch [nginx](#) und [ssl](#).

```
nano /etc/nginx/sites-available/collabora.home
```

```
server {
    listen 80;
#    listen [::]:80;
    server_name collabora.home;

    # enforce https
    return 301 https://$server_name:443$request_uri;
}

server {
    listen 443 ssl http2;
#    listen [::]:443 ssl http2;
    server_name collabora.home;

    ssl_certificate /etc/nginx/ssl/collabora.home.crt;
    ssl_certificate_key /etc/nginx/ssl/collabora.home.key;
    include conf.d/ssl-params.conf;
    access_log /var/log/nginx/collabora.home_access_log;
    error_log /var/log/nginx/collabora.home-error_log;

    location / {
        proxy_pass https://127.0.0.1:9980;
        proxy_http_version 1.1;
        proxy_read_timeout 3600s;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
        add_header X-Frontend-Host $host;
        add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains" always;
    }
}
```

```
ln -s /etc/nginx/sites-available/collabora.home /etc/nginx/sites-enabled/
systemctl reload nginx.service
```

From:

<http://wiki.techsaviours.org/> - **Your Digital Privacy DIY Solutions | TECH SAVIOURS .ORG**

Permanent link:

<http://wiki.techsaviours.org/de/server/services/nextcloud?rev=1680201960>

Last update: **2023/03/30 18:46**

