

```
sudo apt install nginx
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl status nginx
```

```
sudo apt install php7.4 php7.4-fpm php7.4-mysql php-common php7.4-cli php7.4-common php7.4-
json php7.4-opcache php7.4-readline php7.4-mbstring php7.4-xml php7.4-gd php7.4-curl php-
imagick php7.4-zip php7.4-xml php7.4-bz2 php7.4-intl php7.4-bcmath php7.4-gmp
```

```
sudo systemctl start php7.4-fpm
sudo systemctl enable php7.4-fpm
systemctl status php7.4-fpm
```

```
sudo apt install mariadb-server mariadb-client
sudo systemctl status mariadb
```

```
sudo mysql_secure_installation
```

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

```
CREATE DATABASE nextcloud;
CREATE USER 'nc'@'localhost' IDENTIFIED BY 'Passwort';
GRANT ALL PRIVILEGES ON nextcloud.* TO 'nc'@'localhost';
FLUSH PRIVILEGES;
exit
```

```
sudo nano etc/nginx/conf.d/nextcloud.conf
```

```
server {
    listen 80;
    listen [::]:80;
    server_name localhost;

    # Add headers to serve security related headers
    add_header X-Content-Type-Options nosniff;
    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;
    add_header Referrer-Policy no-referrer;
    add_header X-Frame-Options "SAMEORIGIN";

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

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

    location = /robots.txt {
        allow all;
```

```
log_not_found off;
access_log off;
}
```

```
# The following 2 rules are only needed for the user_webfinger app.
# Uncomment it if you're planning to use this app.
#rewrite ^/.well-known/host-meta /public.php?service=host-meta last;
#rewrite ^/.well-known/host-meta.json /public.php?service=host-meta-json
# last;
```

```
location = /.well-known/carddav {
return 301 $scheme://$host/remote.php/dav;
}
location = /.well-known/caldav {
return 301 $scheme://$host/remote.php/dav;
}
```

```
location ~ /.well-known/acme-challenge {
allow all;
}
```

```
# set max upload size
client_max_body_size 512M;
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 /core/templates/403.php;
error_page 404 /core/templates/404.php;
```

```
location / {
rewrite ^ /index.php;
}
```

```
location ~ ^/(?build|tests|config|lib|3rdparty|templates|data)/ {
deny all;
}
location ~ ^/(?:\.|autotest|occ|issue|indie|db_|console) {
deny all;
}
```

```
location ~ ^/(?index|remote|public|cron|core/ajax/update|status|ocs/v[12]|updater/.+|ocs-
provider/.+|core/templates/40[34])\.php(?:$|/) {
include fastcgi_params;
fastcgi_split_path_info ^(.+\.php)(/.*)$;
try_files $fastcgi_script_name =404;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
```

```

fastcgi_param PATH_INFO $fastcgi_path_info;
#Avoid sending the security headers twice
fastcgi_param modHeadersAvailable true;
fastcgi_param front_controller_active true;
fastcgi_pass unix:/run/php/php7.4-fpm.sock;
fastcgi_intercept_errors on;
fastcgi_request_buffering off;
}

```

```

location ~ ^/(?:(?:updater|ocs-provider)(?:$|/)) {
try_files $uri/ =404;
index index.php;
}

```

```

# Adding the cache control header for js and css files
# Make sure it is BELOW the PHP block
location ~* \.(?:css|js)$ {
try_files $uri /index.php$uri$is_args$args;
add_header Cache-Control "public, max-age=7200";
# Add headers to serve security related headers (It is intended to
# have those duplicated to the ones above)
add_header X-Content-Type-Options nosniff;
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;
add_header Referrer-Policy no-referrer;
# Optional: Don't log access to assets
access_log off;
}

```

```

location ~* \.(?:svg|gif|png|html|ttf|woff|ico|jpg|jpeg)$ {
try_files $uri /index.php$uri$is_args$args;
# Optional: Don't log access to other assets
access_log off;
}
}

```

sudo nginx -t

```

sudo apt install certbot -y
systemctl stop nginx
certbot certonly --standalone -d cloud.gw-county.de

```

```

wget https://download.nextcloud.com/server/releases/nextcloud-20.0.7.zip
???sudo unzip nextcloud-21.0.0.zip -d /usr/share/nginx/
???sudo chown -R www-data:www-data /usr/share/nginx/nextcloud
???sudo unzip nextcloud-21.0.0.zip -d /var/www/nextcloud
???sudo chown -R www-data:www-data //var/www/nextcloud

```

VHost anlegen:

```
cd /etc/nginx/sites-available/  
nano nextcloud
```

```
upstream php-handler {  
    #server 127.0.0.1:9000;  
    server unix:/var/run/php/php7.4-fpm.sock;  
}
```

```
server {  
    listen 80;  
    listen [::]:80;  
    server_name cloud.gw-county.de;  
    return 301 https://$server_name:443$request_uri;  
}
```

```
server {  
    listen 443 ssl http2;  
    listen [::]:443 ssl http2;  
    server_name cloud.gw-county.de;  
    ssl_certificate /etc/letsencrypt/live/cloud.gw-county.de/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/cloud.gw-county.de/privkey.pem;  
    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 "none" always;  
    add_header X-XSS-Protection "1; mode=block" always;  
    fastcgi_hide_header X-Powered-By;  
    root /var/www/nextcloud;  
    location = /robots.txt {  
        allow all;  
        log_not_found off;  
        access_log off;  
    }  
    location = /.well-known/carddav {  
        return 301 $scheme://$host:$server_port/remote.php/dav;  
    }  
    location = /.well-known/caldav {  
        return 301 $scheme://$host:$server_port/remote.php/dav;  
    }  
    client_max_body_size 512M;  
    fastcgi_buffers 64 4K;  
    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/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;

```
location / {
rewrite ^ /index.php;
}
```

```
location ~ ^\(?:build|tests|config|lib|3rdparty|templates|data\) {
deny all;
}
location ~ ^\(?:\.|autotest|occ|issue|indie|db_|console) {
deny all;
}
```

```
location ~ ^\(?:index|remote|public|cron|core|vajax|update|status|ocs|v[12]|updater|.+|oc[ms]-provider|.+)\.php(?:$|\/) {
fastcgi_split_path_info ^(.+?\.php)(\/.*|)$;
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 ~ ^\(?:updater|oc[ms]-provider)(?:$|\/) {
try_files $uri/ =404;
index index.php;
}
```

```
location ~ \.(?:css|js|woff2?|svg|gif|map)$ {
try_files $uri /index.php$request_uri;
add_header Cache-Control "public, max-age=15778463";
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 "none" always;
add_header X-XSS-Protection "1; mode=block" always;
access_log off;
}
```

```
location ~ \.(?:png|html|ttf|ico|jpg|jpeg|bcmap)$ {  
    try_files $uri /index.php$request_uri;  
    access_log off;  
}  
}
```

VHost aktivieren:

```
ln -s /etc/nginx/sites-available/nextcloud /etc/nginx/sites-enabled/  
nginx -t
```

```
systemctl restart nginx  
systemctl restart php7.4-fpm
```

Nextcloud abschließende Konfiguration:

Aufruf: <http://cloud.gw.county.de>

Hier sollte direkt eine Umleitung auf https erfolgen, anschließend die Konfiguration durchführen

make sure to add the port number 3306 by localhost. For example, localhost:3306, that is the port number MariaDB listens on.

How to Move the Data Directory

In case you need to move the NextCloud data directory, there are 4 steps to accomplish this. First, you need to use the `cp` command to copy the data directory to the new directory. For example, the mount point of my external hard drive is `/media/linuxbabe/b43e4eea-9796-4ac6-9c48-2bcaa46353731`. I create the new data directory on the external hard drive.

```
sudo mkdir /media/linuxbabe/b43e4eea-9796-4ac6-9c48-2bcaa46353731/nextcloud-data/
```

Then I copy the original data directory to the new data directory. `-R` flag means the copy operation is recursive.

```
sudo cp /usr/share/nginx/nextcloud-data/* /media/linuxbabe/b43e4eea-9796-4ac6-9c48-2bcaa46353731/nextcloud-data/ -R
```

You also need to copy the `.ocdata` file.

```
sudo cp /usr/share/nginx/nextcloud-data/.ocdata /media/linuxbabe/b43e4eea-9796-4ac6-9c48-2bcaa46353731/nextcloud-data/
```

Next, you need to set `www-data` (Nginx user) as the owner.

```
sudo chown www-data:www-data /media/linuxbabe/b43e4eea-9796-4ac6-9c48-2bcaa46353731/nextcloud-data/ -R
```

Lastly, you need to edit the `config.php` file.

```
sudo nano /usr/share/nginx/nextcloud/config/config.php
```

Find the following line and change the value of `datadirectory`.

```
'datadirectory' => '/usr/share/nginx/nextcloud-data',
```

Save and close the file. Reload NextCloud web page and you are done.

Step 7: Increase PHP Memory Limit

The default PHP memory limit is 128MB. NextCloud recommends 512MB for better performance. To change PHP memory limit, edit the `php.ini` file.

```
sudo nano /etc/php/7.4/fpm/php.ini
```

Find the following line. (line 409)

```
memory_limit = 128M
```

Change the value.

```
memory_limit = 512M
```

Save and close the file. Alternatively, you can run the following command to change the value without manually opening the file.

```
sudo sed -i 's/memory_limit = 128M/memory_limit = 512M/g' /etc/php/7.4/fpm/php.ini
```

Then reload PHP-FPM service for the changes to take effect.

```
sudo systemctl reload php7.4-fpm
```

Step 8: Set Up PHP to Properly Query System Environment Variables

Edit the `www.conf` file.

```
sudo nano /etc/php/7.4/fpm/pool.d/www.conf
```

Find the following line (line 396).

```
;clear_env = no
```

Remove the semicolon to uncomment this line.

```
clear_env = no
```

Save and close the file. Alternatively, you can run the following command to uncomment this line without manually opening the file.

```
sudo sed -i 's/clear_env = no/clear_env = no/g' /etc/php/7.4/fpm/pool.d/www.conf
```

Then reload PHP-FPM service for the changes to take effect.

```
sudo systemctl reload php7.4-fpm
```

Step 9: Increase Upload File Size Limit

The default maximum upload file size limit set by Nginx is 1MB. To allow uploading large files to your NextCloud server, edit the Nginx configuration file for NextCloud.

```
sudo nano /etc/nginx/conf.d/nextcloud.conf
```

We have already set the maximum file size in this file, as indicated by

```
client_max_body_size 512M;
```

You can change it if you prefer, like 1G.

```
client_max_body_size 1024M;
```

Save and close the file. Then reload Nginx for the changes to take effect.

```
sudo systemctl reload nginx
```

PHP also sets a limit of upload file size. The default maximum file size for uploading is 2MB. To increase the upload size limit, edit the PHP configuration file.

```
sudo nano /etc/php/7.4/fpm/php.ini
```

Find the following line (line 846).

```
upload_max_filesize = 2M
```

Change the value like below:

```
upload_max_filesize = 1024M
```

Save and close the file. Alternatively, you can run the following command to change the value without manually opening the file.

```
sudo sed -i 's/upload_max_filesize = 2M/upload_max_filesize = 1024M/g' /etc/php/7.4/fpm/php.ini
```

Then restart PHP-FPM.

```
sudo systemctl restart php7.4-fpm
```

Step 10: Configure Redis Cache for NextCloud

If you go to your NextCloud settings -> overview page, you might see the following warning:

No memory cache has been configured. To enhance your performance please configure a memcache if available.

We will enable memory caching for nextCloud by using Redis. Run the following command to install Redis server from Ubuntu repository.

```
sudo apt install redis-server
```

You can check the version with:

```
redis-server -v
```

Sample output:

```
Redis server v=5.0.7 sha=00000000:0 malloc=jemalloc-5.2.1 bits=64  
build=636cde3b5c7a3923
```

Now we can check if redis server is running.

```
systemctl status redis
```

```
linuxbabe@focal:~$ systemctl status redis  
● redis-server.service - Advanced key-value store  
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: enabled)  
   Active: active (running) since Wed 2020-04-29 20:25:34 HKT; 18s ago  
     Docs: http://redis.io/documentation,  
           man:redis-server(1)  
  Process: 3686487 ExecStart=/usr/bin/redis-server /etc/redis/redis.conf (code=exited, status=0/  
Main PID: 3686505 (redis-server)  
   Tasks: 4 (limit: 9451)  
  Memory: 2.3M  
   CGroup: /system.slice/redis-server.service  
           └─3686505 /usr/bin/redis-server 127.0.0.1:6379
```

Hint: If the above command didn't quit immediately, you can press the Q key to gain back control of the terminal.

From the above screenshot, we can see that it's running and auto-start is enabled. If for any reason it's not running, execute the following command:

```
sudo systemctl start redis-server
```

And if auto-start at boot time is not enabled, you can use the following command to enable it:

```
sudo systemctl enable redis-server
```

In order to configure Redis as a cache for nextCloud, we need to install the PHP extension for interfacing with Redis.

```
sudo apt install php-redis
```

Check if the extension is enabled.

```
php --ri redis
```

```
linuxbabegfocal:~$ php --ri redis
redis
Redis Support => enabled
Redis Version => 5.1.1
Available serializers => php, json, igbinary
```

We can see that Redis extension is enabled. If it's not enabled, run the following command:

```
sudo phpenmod redis
```

Next, edit nextCloud configuration file.

```
sudo nano /usr/share/nginx/nextcloud/config/config.php
```

Add the following lines above the ending `);` line.

```
'memcache.distributed' => '\OC\Memcache\Redis',
'memcache.local' => '\OC\Memcache\Redis',
'memcache.locking' => '\OC\Memcache\Redis',
'redis' => array(
    'host' => 'localhost',
    'port' => 6379,
),
```

```
'mysql.utf8mb4' => true,
'dbuser' => 'nextclouduser',
'dbpassword' => ' ',
'installed' => true,

'memcache.distributed' => '\OC\Memcache\Redis',
'memcache.local' => '\OC\Memcache\Redis',
'memcache.locking' => '\OC\Memcache\Redis',
'redis' => array(
    'host' => 'localhost',
    'port' => 6379,
),
);
```

Save and close the file. Then restart Nginx and PHP-FPM.

```
sudo systemctl restart nginx php7.4-fpm
```

Now go to NextCloud settings -> overview page again and refresh the web page, the warning about memory caching should be gone.

Adding Missing Indexes

If you see the following message in the NextCloud Settings -> Overview page,

The database is missing some indexes. Due to the fact that adding indexes on big tables could take some time they were not added automatically.

Then you need to manually add those indexes. Change to the Nextcloud webroot directory.

```
cd /usr/share/nginx/nextcloud/
```

Run the following command to add indexes to the Nextcloud database.

```
sudo -u www-data php occ db:add-missing-indices
```

```
linuxbabe@focal:/usr/share/nginx/nextcloud$ sudo -u www-data php occ db:add-missing-indices
Check indices of the share table.
Check indices of the filecache table.
Check indices of the twofactor_providers table.
Check indices of the login_flow_v2 table.
Check indices of the whats_new table.
Check indices of the cards table.
Check indices of the cards_properties table.
Check indices of the calendarobjects_props table.
Adding calendarobject_calid_index index to the calendarobjects_props table, this can take some time
...
calendarobjects_props table updated successfully.
Check indices of the schedulingobjects table.
Adding schedulobj_principuri_index index to the schedulingobjects table, this can take some time...
schedulingobjects table updated successfully.
```

Now if you refresh the NextCloud Settings -> Overview page, the warning about missing indexes should be gone.

Conversion to Big Int

If you see the following message in the NextCloud Settings -> Overview page,

Some columns in the database are missing a conversion to big int. Due to the fact that changing column types on big tables could take some time they were not changed automatically.

Then you need to manually change the column type. Change to the Nextcloud webroot directory.

```
cd /usr/share/nginx/nextcloud/
```

Change your Nextcloud into maintenance mode to prevent users from logging in and making changes.

```
sudo -u www-data php occ maintenance:mode --on
```

Then run the following command to change the column type.

```
sudo -u www-data php occ db:convert-filecache-bigint
```

Once it's done, switch off the maintenance mode.

```
sudo -u www-data php occ maintenance:mode --off
```

```

linuxbabe@focal:/usr/share/nginx/nextcloud$ sudo -u www-data php occ maintenance:mode --on
Maintenance mode enabled
linuxbabe@focal:/usr/share/nginx/nextcloud$ sudo -u www-data php occ db:convert-filecache-bigint
Nextcloud is in maintenance mode - no apps have been loaded

Following columns will be updated:

* mounts.storage_id
* mounts.root_id
* mounts.mount_id

This can take up to hours, depending on the number of files in your instance!
Continue with the conversion (y/n)? [n] y
linuxbabe@focal:/usr/share/nginx/nextcloud$ sudo -u www-data php occ maintenance:mode --off
Maintenance mode disabled
linuxbabe@focal:/usr/share/nginx/nextcloud$

```

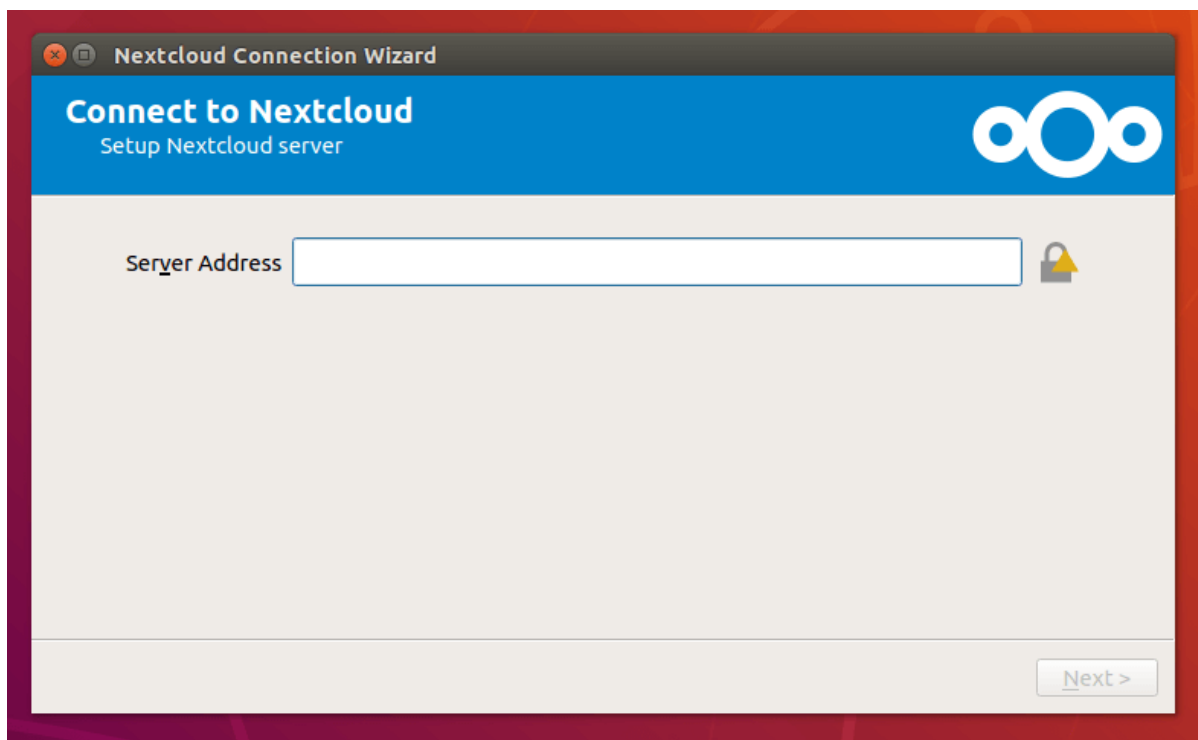
Now if you refresh the NextCloud Settings -> Overview page, the warning about big int should be gone.

How to Install NextCloud Client on Ubuntu 20.04 Desktop

Run the following commands on Ubuntu 20.04 desktop to install the client from the default repository.

```
sudo apt install nextcloud-client
```

NextCloud Client on Ubuntu 20.04

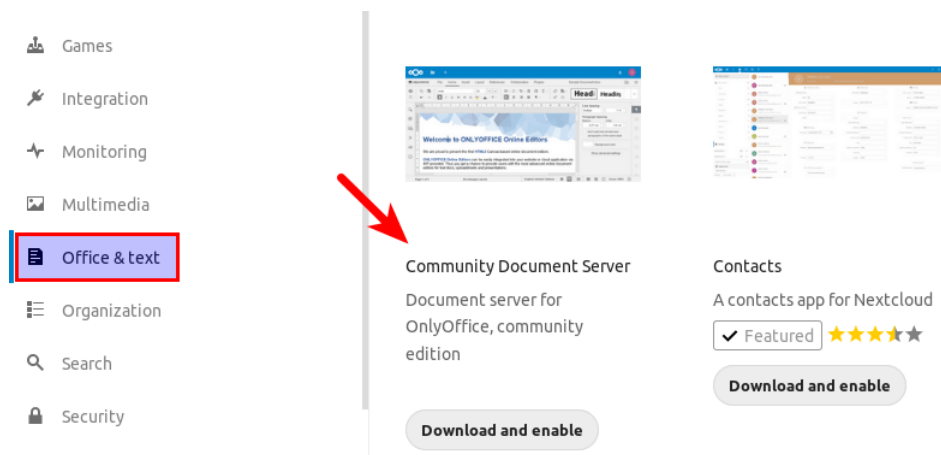


Client software for macOS, Windows, Android and iOS can be found on the [Nextcloud download page](#).

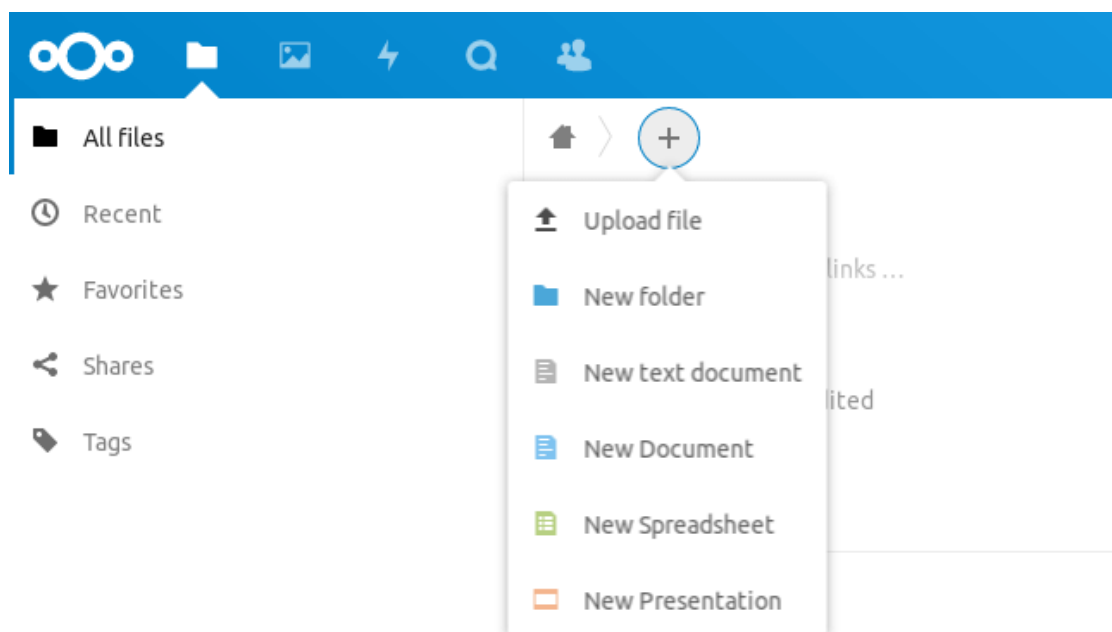
How to Enable OnlyOffice/Collabora Online

By default, Nextcloud ships with support for OnlyOffice, which is an online office suite that allows you to edit your doc, ppt, xls files directly from NextCloud. We only need to install an app to use

this feature. Go to Nextcloud Apps -> Office & Text. Find and enable the community document server app.



Now when you click the add button (+) in Nextcloud, you will be able to create Word, spreadsheet and presentation documents right from your Nextcloud server.



However, I found this app isn't very reliable. And the community edition allows only 20 users at most. You need to purchase an enterprise edition if you have more than 20 users. There's another open-source LibreOffice-based online office suite called Collabora Online that has the same functionality, but without the limitation on the number of users. You can read the following article to integrate it with Nextcloud.

- [Integrate Collabora Online with Nextcloud on Ubuntu with Docker](#)

Adding Local DNS Entry

It's recommended to edit the `/etc/hosts` file on your Nextcloud server and add the following entry, so that Nextcloud itself won't have to query the public DNS, which can improve the overall

stability. If your Nextcloud server can't resolve the `nextcloud.example.com` hostname, then you may encounter a 504 gateway time out error.

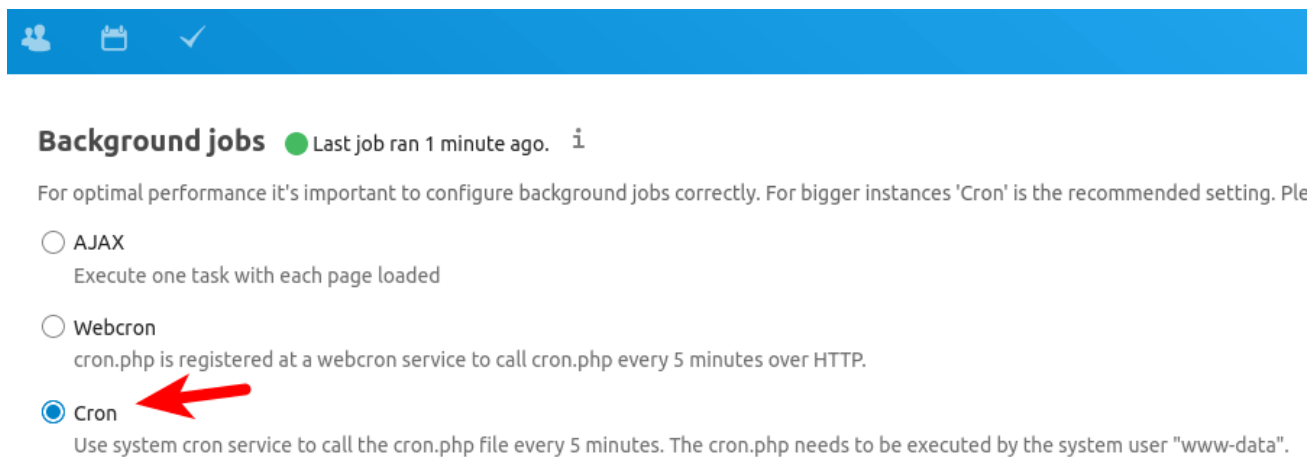
```
127.0.0.1 localhost nextcloud.example.com
```

An IP address in the `/etc/hosts` file can have multiple hostnames, so if you have other applications installed on the same box, you can also add other hostnames or sub-domains on the same line like this:

```
127.0.0.1 localhost focal ubuntu nextcloud.example.com collabora.example.com
```

Using Cron to Run Background Jobs

By default, Nextcloud uses AJAX to execute one task with each page load. You can use the more efficient system cron service to run background jobs. Go to Nextcloud Settings -> Basic Settings and select Cron.



Next, edit the `www-data` user's crontab file.

```
sudo -u www-data crontab -e
```

Add the following line in this file, so the cron job will run every 5 minutes.

```
*/* * * * * php7.4 -f /usr/share/nginx/nextcloud/cron.php
```

Save and close the file.

(Optional) Prevent Malicious Login Attempts

If your computer has a static public IP address, you can create an IP whitelist in the Nginx config file.

```
sudo nano /etc/nginx/conf.d/nextcloud.conf
```

Add the following lines in the SSL server block to restrict access to the `/login` URL, so only your IP address can access this URL. Replace 78.56.34.12 with your own IP address.

```
location ~* ^/login{
    try_files $uri /index.php;
```

```

include fastcgi_params;
fastcgi_split_path_info ^(.+\.php)(/.*)$;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
fastcgi_param PATH_INFO $fastcgi_path_info;
#Avoid sending the security headers twice
fastcgi_param modHeadersAvailable true;
fastcgi_param front_controller_active true;
fastcgi_pass unix:/run/php/php7.4-fpm.sock;
fastcgi_intercept_errors on;
fastcgi_request_buffering off;

allow 78.56.34.12;
deny all;
}

```

Save and close the file. Then test Nginx configuration.

```
sudo nginx -t
```

If the test is successful, reload Nginx for the changes to take effect.

```
sudo systemctl reload nginx
```

If you don't have a static IP address on your home network, you can [set up a VPN server in a data center](#).

Troubleshooting Tips

If you encounter errors, you can check one of the following log files to find out what's wrong.

- Nginx error log: `/var/log/nginx/error.log`
- Nginx error log for the Nextcloud virtual host: `/var/log/nginx/nextcloud.error`
- Nextcloud application log: `/usr/share/nginx/nextcloud/data/nextcloud.log`

For example, I once had an "Internal Server Error" on my Nextcloud instance and the `/var/log/nginx/nextcloud.error` file told me that

```
FastCGI sent in stderr: "PHP message: PHP Fatal error: Uncaught OC\HintException: [0]: Memcache \OC\Memcache\Redis not available for local cache (Is the matching PHP module installed and enabled?)"
```

It turned out that because I used [the ppa:ondrej/php PPA](#) on my Ubuntu server, I also need to install `php7.4-redis` in addition to the `php-redis` package.

Upgrading Nextcloud

It's important to keep your Nextcloud server up to date with the latest security and bug fixes. Read the tutorial below to learn how to upgrade Nextcloud.

- [2 Ways to Upgrade Nextcloud \[Command-Line & GUI\]](#)