nextcloud

- □□□□

□□□□

android □□CardDav CALDAV□□

1. □□f-droid
2. □f-droid□□davx5
3. □□tasks

□user□□□□□□□□□□disable□

□□1 □□□□

```
UPDATE oc_preferences SET configvalue = 'true' WHERE appid = 'core' AND configkey = 'enabled' AND userid = 'user_id'
```

□□2 php□□

```
sudo -u www-data php occ user:enable user_id
```

□□locked 423□□

□□□config.php□□'filelockinf.enabled'=>false,

□□□□

1. config.php 'maintance'=>ture;
2. Delete from oc_file_locks where 1;
3. config.php 'maintance'=>false;

□□□□local nolyoffice server□□□

□□□Error when trying to connect (Host violates local access rules)□□□□□
vi config/config.php

```
'allow_local_remote_servers' => true,    □□□□
```

□□□□□□□□□□□□□□□

□□app customer CSS□□□□

```
#contactsmenu {
	display: none;
}
```

□□□□□□□□□□□□timeout

1□php.ini □□“upload_max_filesize” “post_max_size” and “memory_limit” □□□□
2□nginx□□client_max_body_size□client_body_timeout□fastcgi_buffers □□□□
3□clamav□□□□□□□□□□

cron.php □□□□□

□□□□□nextcloud□□□□-□□□□□□□□□□□□□□□□□□□□□□□xx□□
□□cmd□□sudo -u www-data php -f □□/cron.php □□□□□□□
PHP Fatal error: Allowed memory size of 2097152 bytes exhausted (tried to allocate 438272 bytes) in /home/ronfs/nextcloud/fs.dreamteenet.com/3rdparty/composer/autoload_real.php on line 37 □□□
□php.ini □memory_limit □□-1□□□□□□

□□carddav□□

□□□□□https://cloud.nextcloud.com/remote.php/dav/calendars/daniel/personal/

□□webdav□□

□□□□□davs://example.com/nextcloud/remote.php/dav/files/USERNAME/

https://cloud.YOURDOMAIN.com/remote.php/dav/files/USERNAME/

windows:

```
net use Z: https://example.com/nextcloud/remote.php/dav/files/USERNAME/ /user:youruser yourpassword
```

Since WebDAV is an extension of HTTP, cURL can be used to script file operations.

Note
Settings → Administration → Sharing → Allow users on this server to send shares to other servers.
 If this option is disabled the option --header "X-Requested-With: XMLHttpRequest" needs to be passed to cURL.

To create a folder with the current date as name:
$ curl -u user:pass -X MKCOL
"https://example.com/nextcloud/remote.php/dav/files/USERNAME/$(date '+%d-%b-%Y')"

To upload a file error.log into that directory:
$ curl -u user:pass -T error.log
"https://example.com/nextcloud/remote.php/dav/files/USERNAME/$(date '+%d-%b-%Y')/error.log"

To move a file:
$ curl -u user:pass -X MOVE --header 'Destination:
https://example.com/nextcloud/remote.php/dav/files/USERNAME/target.jpg'
https://example.com/nextcloud/remote.php/dav/files/USERNAME/source.jpg

backup & restore

backup

maintenance mode
way1

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

way2
You may also put your server into this mode by editing config/config.php. Change "maintenance" => false to "maintenance" => true:

```
<?php
 "maintenance" => true,
```

Don’t forget to change it back to false when you are finished.

folders backup

```
rsync -Aavx nextcloud/ nextcloud-dirbkp_`date +"%Y%m%d"`/
```

Database backup

MySQL/MariaDB
MySQL or MariaDB, which is a drop-in MySQL replacement, is the recommended database engine. To backup MySQL/MariaDB:

```
mysqldump --single-transaction -h [server] -u [username] -p[password] [db_name] > nextcloud-sqlbkp_`date +"%Y%m%d"`.bak
```

If you use enabled MySQL/MariaDB 4-byte support (Enabling MySQL 4-byte support, needed for emoji), you will need to add --default-character-set=utf8mb4 like this:

```
mysqldump --single-transaction --default-character-set=utf8mb4 -h [server] -u [username] -p[password] [db_name] > nextcloud-sqlbkp_`date +"%Y%m%d"`.bak
```

SQLite

```
sqlite3 data/owncloud.db .dump > nextcloud-sqlbkp_`date +"%Y%m%d"`.bak
```

PostgreSQL

```
PGPASSWORD="password" pg_dump [db_name] -h [server] -U [username] -f nextcloud-sqlbkp_`date +"%Y%m%d"`.bak
```

restore

folder restore

```
rsync -Aax nextcloud-dirbkp/ nextcloud/
```

database restore

Warning

Before restoring a backup you need to make sure to delete all existing database tables.
The easiest way to do this is to drop and recreate the database. SQLite does this automatically.
MySQL
MySQL is the recommended database engine. To restore MySQL:

```
mysql -h [server] -u [username] -p[password] -e "DROP DATABASE nextcloud"
mysql -h [server] -u [username] -p[password] -e "CREATE DATABASE nextcloud"
```

If you use UTF8 with multibyte support (e.g. for emojis in filenames), use:

```
mysql -h [server] -u [username] -p[password] -e "DROP DATABASE nextcloud"
mysql -h [server] -u [username] -p[password] -e "CREATE DATABASE nextcloud CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci"
```

PostgreSQL

```
PGPASSWORD="password" psql -h [server] -U [username] -d template1 -c "DROP DATABASE \"nextcloud\";"
PGPASSWORD="password" psql -h [server] -U [username] -d template1 -c "CREATE DATABASE \"nextcloud\";"
```

Restoring

Note
This guide assumes that your previous backup is called “nextcloud-sqlbkp.bak”
MySQL

MySQL is the recommended database engine. To restore MySQL:

```
mysql -h [server] -u [username] -p[password] [db_name] < nextcloud-sqlbkp.bak
```

SQLite

```
rm data/owncloud.db
sqlite3 data/owncloud.db < nextcloud-sqlbkp.bak
```

PostgreSQL

```
PGPASSWORD="password" psql -h [server] -U [username] -d nextcloud -f nextcloud-sqlbkp.bak
```

Maintenance commands

Use these commands when you upgrade Nextcloud, manage encryption, perform backups and other tasks that require locking users out until you are finished:

maintenance

```
maintenance:data-fingerprint        update the systems data-fingerprint after a backup is restored
maintenance:mimetype:update-db      Update database mimetypes and update filecache
maintenance:mimetype:update-js      Update mimetypelist.js
maintenance:mode                    set maintenance mode
maintenance:repair                  repair this installation
maintenance:theme:update            Apply custom theme changes
maintenance:update:htaccess         Updates the .htaccess file
```

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

```
sudo -u www-data php occ maintenance:repair
```

□□□Nextcloud□□□□□□

```
'filesystem_check_changes' => 1   #□config.php□□
php occ files:scan --all          #□□□□□□□□
```

□□ occ □□□□ oc_filecache

```
sudo -u www-data php occ files:scan --all
```

nextcloud□□□□□□□□□□□

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

□□□□□□

```
'trashbin_retention_obligation' => 'auto',      #□□□□config.php□□□□□□□□auto
```

default setting. keeps files and folders in the trash bin for 30 days and automatically deletes anytime after that if space is needed (note: files may not be deleted if space is not needed).
D, auto
keeps files and folders in the trash bin for D+ days, delete anytime if space needed (note: files may not be deleted if space is not needed)
auto, D
delete all files in the trash bin that are older than D days automatically, delete other files anytime if space needed
D1, D2
keep files and folders in the trash bin for at least D1 days and delete when exceeds D2 days (note: files will not be deleted automatically if space is needed)
disabled
trash bin auto clean disabled, files and folders will be kept forever

□□□□
Deactivate background job:

```
occ config:app:set --value=no files_trashbin background_job_expire_trash
```

Activate background job:

```
occ config:app:delete files_trashbin background_job_expire_trash
```

□□ip(□□□□□□□□ip□□□)

```
DELETE FROM oc_bruteforce_attempts WHERE ip = "x.x.x.x";
```

□□□□□□□□□

```
<?php
 'preview_format' => 'webp',
```

□□□□□□□□□80□

```
occ config:app:set preview webp_quality --value="30"
```

□□APCu□Redis□□□□

```
'memcache.local' => '\OC\Memcache\APCu',
'memcache.distributed' => '\OC\Memcache\Redis',
'memcache.locking' => '\OC\Memcache\Redis',
'redis' => [
     'host' => 'localhost',
     'port' => 6379,
],
```