

Encryption commands



- [Status and control](#)
 - [encryption:status](#)
 - [encryption:enable](#)
 - [encryption:disable](#)
- [Encrypt and decrypt all data](#)
 - [encryption:encrypt-all](#)
 - [encryption:decrypt-all](#)
- [Encryption modules](#)
 - [encryption:list-modules](#)
 - [encryption:set-default-module](#)
- [Key storage](#)
 - [encryption:show-key-storage-root](#)
 - [encryption:change-key-storage-root](#)
 - [encryption:migrate-key-storage-format](#)
- [Master key](#)
 - [encryption:enable-master-key](#)
 - [encryption:disable-master-key](#)
- [Maintenance and repair](#)
 - [encryption:clean-orphaned-keys](#)
 - [encryption:fix-encrypted-version](#)
 - [encryption:fix-key-location](#)
 - [encryption:recover-user](#)
- [Legacy migration](#)
 - [encryption:drop-legacy-filekey](#)
 - [encryption:scan:legacy-format](#)

The `encryption` commands manage server-side encryption, encryption keys, and encryption modules. The core commands are always available. Commands that operate on the encryption module itself (master key, key repair, legacy migration) require the **Encryption** app (`encryption`)

to be enabled.



For a full guide to configuring server-side encryption, see [../configuration files/encryption configuration](https://docs.aws.amazon.com/iam/latest/userguide/encryption-configuration.html).

encryption	
encryption:change-key-storage-root	change key storage root
encryption:clean-orphaned-keys	scan the keys storage for orphaned keys and remove them
encryption:decrypt-all	disable server-side encryption and decrypt all files
encryption:disable	disable encryption
encryption:disable-master-key	disable the master key and use per-user keys instead
encryption:drop-legacy-filekey	scan the files for the legacy filekey format using RC4 and get rid of it
encryption:enable	enable encryption
encryption:enable-master-key	enable the master key
encryption:encrypt-all	encrypt all files for all users
encryption:fix-encrypted-version	fix the encrypted version if the encrypted file(s) are not downloadable
encryption:fix-key-location	fix the location of encryption keys for external storage
encryption:list-modules	list all available encryption modules
encryption:migrate-key-storage-format	migrate the format of the key storage to a newer format
encryption:recover-user	recover user data in case of password loss
encryption:scan:legacy-format	scan the files for the legacy format
encryption:set-default-module	set the encryption default module
encryption:show-key-storage-root	show current key storage root
encryption:status	lists the current status of encryption

Status and control

encryption:status

Show whether encryption is enabled and which module is active:

```
sudo -E -u www-data php occ encryption:status
- enabled: false
- defaultModule: OC_DEFAULT_MODULE
```

Use `--output=json_pretty` for machine-readable output.

encryption:enable

Enable server-side encryption. The **Encryption** app must be enabled first and a default module must be configured:

```
sudo -E -u www-data php occ app:enable encryption
sudo -E -u www-data php occ encryption:enable
Encryption enabled

Default module: OC_DEFAULT_MODULE
```

encryption:disable

Disable server-side encryption. This only disables the encryption flag — it does not decrypt any files:

```
sudo -E -u www-data php occ encryption:disable
Encryption disabled
```

To also decrypt all existing files, run `encryption:decrypt-all` afterwards.

Encrypt and decrypt all data

encryption:encrypt-all

Encrypt all files for all users. Encryption must be enabled before running this command. The command manages maintenance mode internally — do **not** enable maintenance mode first, as the command will fail if it is already active:

```
sudo -E -u www-data php occ encryption:encrypt-all
```

You are about to encrypt all files stored in your Nextcloud installation.

Depending on the number of available files, and their size, this may take quite some time.

Please ensure that no user accesses their files during this time!

Note: The encryption module you use determines which files get encrypted.

Do you really want to continue? (y/n)

The command requires an interactive terminal (TTY). If running inside a Docker container, use `docker exec -it`.

encryption:decrypt-all

Decrypt all files for all users, or for a single user. The command manages maintenance mode internally — do **not** enable maintenance mode first:

```
sudo -E -u www-data php occ encryption:decrypt-all
```

```
sudo -E -u www-data php occ encryption:decrypt-all layla
```

When decrypting all users, server-side encryption is disabled automatically. When decrypting a single user, encryption remains enabled for others.

Depending on the encryption module in use, decryption may require:

- **Master key mode** — no extra credentials needed.
- **Per-user key mode** — the user must have enabled the recovery key on their personal settings page.

The command requires an interactive terminal (TTY). If running inside a Docker container, use `docker exec -it`.

Encryption modules

encryption:list-modules

List all available encryption modules. The active default module is marked with `[default*]`:

```
sudo -E -u www-data php occ encryption:list-modules
```

```
- OC_DEFAULT_MODULE: Default encryption module [default*]
```

encryption:set-default-module

Set the default encryption module. Maintenance mode must be disabled:

```
sudo -E -u www-data php occ encryption:set-default-module OC_DEFAULT_MODULE  
Set default module to "OC_DEFAULT_MODULE"
```

Key storage

encryption:show-key-storage-root

Show where encryption keys are stored:

```
sudo -E -u www-data php occ encryption:show-key-storage-root  
Current key storage root:  default storage location (data/)
```

encryption:change-key-storage-root

Move encryption keys to a different directory. The target directory must exist and be writable by the web server user before running the command:

```
sudo -E -u www-data php occ encryption:change-key-storage-root /etc/nextcloud/keys  
Change key storage root from default storage location to /etc/nextcloud/keys  
Start to move keys:  
[=====]  
Key storage root successfully changed to /etc/nextcloud/keys
```

Omit the argument to reset the key storage root back to the default location (`data/`). The command will ask for confirmation:

```
sudo -E -u www-data php occ encryption:change-key-storage-root  
No storage root given, do you want to reset the key storage root to the default location?  
(y/n)
```

encryption:migrate-key-storage-format

Migrate the key storage to the current format (JSON-wrapped, re-encrypted with the server secret). Run this once after upgrading from an installation that used the legacy key format:

```
sudo -E -u www-data php occ encryption:migrate-key-storage-format
Updating key storage format
Start to update the keys:
[=====]
Key storage format successfully updated
```

Master key

The master key encrypts all user data with a single server-managed key. It is the recommended setup for new installations because it simplifies key management and enables admin-side decryption without per-user passwords. The master key is enabled by default when the Encryption app is first enabled.

Both commands are only available when the **Encryption** app is enabled.

“ ”

Switching between master key and per-user key mode is only safe on a **fresh installation with no existing encrypted data**. If you switch modes on an instance that already has encrypted files, those files will become permanently inaccessible: the new mode looks for keys that were never created for the existing data. **There is no recovery path**. Always run `encryption:decrypt-all` first if you need to change modes on an existing installation.

“ ”

Despite the warning shown by both commands ("no way to enable/disable it again"), the switch is technically reversible as long as no encrypted data exists yet. The warning is intended to convey that you cannot safely switch modes once users have data.

encryption:enable-master-key

Enable the master key. Only use this on a **fresh installation with no existing encrypted data**. The command prompts for confirmation:

```
sudo -E -u www-data php occ encryption:enable-master-key  
Master key successfully enabled.
```

encryption:disable-master-key

Disable the master key and revert to per-user keys. Only use this on a **fresh installation with no existing encrypted data**. The command prompts for confirmation:

```
sudo -E -u www-data php occ encryption:disable-master-key  
Master key successfully disabled.
```

Switching to per-user keys has the following consequences:

- **Performance** — per-user key operations are slower. Each user's key must be derived individually on every file access.
- **Password loss means data loss** — without the master key there is no admin-side decryption path. Each user must enable the recovery key on their personal settings page *before* losing their password. Without it, `encryption:recover-user` cannot help and the files cannot be recovered.
- **Admin decryption is no longer possible** — admins cannot decrypt or access a user's files without that user's password or a pre-set recovery key.

Maintenance and repair

encryption:clean-orphaned-keys

Scan the key storage for keys that no longer have a corresponding file and remove them. Optionally limit the scan to a single user:

```
sudo -E -u www-data php occ encryption:clean-orphaned-keys  
sudo -E -u www-data php occ encryption:clean-orphaned-keys layla
```

When orphaned keys are found, the command lists them and asks interactively whether to delete all, specific ones, or none:

```
sudo -E -u www-data php occ encryption:clean-orphaned-keys layla
Key storage scan for layla
=====

Orphaned key found: /layla/files_encryption/keys/files/old-doc.pdf/OC_DEFAULT_MODULE/fileKey
Do you want to delete all orphaned keys? (y/n)
```

If no orphaned keys are found, the command exits silently.

encryption:fix-encrypted-version

Fix the encrypted version counter in the file cache when encrypted files cannot be downloaded.
Requires **master key** encryption.

Run for a single user:

```
sudo -E -u www-data php occ encryption:fix-encrypted-version layla
Verifying the content of file "/layla/files/Documents/report.pdf"
The file "/layla/files/Documents/report.pdf" is: OK
```

Run for all users:

```
sudo -E -u www-data php occ encryption:fix-encrypted-version --all
Processing files for layla
Verifying the content of file "/layla/files/Documents/report.pdf"
The file "/layla/files/Documents/report.pdf" is: OK
```

When a broken file is found, the command tries decrementing and then incrementing the encrypted version counter until the file can be read:

```
Attempting to fix the path: "/layla/files/Documents/broken.pdf"
Decrement the encrypted version to 2
Fixed the file: "/layla/files/Documents/broken.pdf" with version 2
```

Use `-p` / `--path` to limit the scan to a specific directory:

```
sudo -E -u www-data php occ encryption:fix-encrypted-version layla --path="/Documents"
```

encryption:fix-key-location

Fix the location of encryption keys for files on external storage mounts. Run this when users cannot access files on external storage after a migration:

```
sudo -E -u www-data php occ encryption:fix-key-location layla  
Migrating key for /layla/files/ExternalDrive/report.pdf ✓
```

Use `--dry-run` to list files that need migration without making any changes:

```
sudo -E -u www-data php occ encryption:fix-key-location --dry-run layla  
/layla/files/ExternalDrive/report.pdf needs migration
```

encryption:recover-user

Recover a user's files after a password loss. Only available in **per-user key** mode (not applicable when master key is enabled). The user must have enabled the recovery key in their personal settings before the password was lost.

The command prompts for the recovery key password and the new login password:

```
sudo -E -u www-data php occ encryption:recover-user layla  
Please enter the recovery key password:  
Please enter the new login password for the user:  
Start to recover users files... This can take some time...Done.
```

Legacy migration

encryption:drop-legacy-filekey

Scan all files for the legacy RC4 filekey format and migrate them to the current format. Requires **master key** encryption. Files not migrated by this command will be migrated automatically on their next write. Running this command upfront also converts old base64-encoded files to binary format, saving approximately 33% disk space:

```
sudo -E -u www-data php occ encryption:drop-legacy-filekey  
Scanning all files for legacy filekey  
Scanning all files for layla  
All scanned files are properly encrypted.
```

encryption:scan:legacy-format

Scan all files for the legacy encryption format. Use this to determine whether any files still need migration before disabling legacy format compatibility:

```
sudo -E -u www-data php occ encryption:scan:legacy-format  
Scanning all files for legacy encryption  
Scanning all files for layla  
All scanned files are properly encrypted. You can disable the legacy compatibility mode.
```

If files using the legacy format are found, their paths are listed. Run `encryption:drop-legacy-filekey` to migrate them.

```
### #2  
[ ] cocobook-sync [ ] 2026-10-28 16:27:22 UTC  
[ ] cocobook-sync [ ] 2026-10-29 01:09:17 UTC
```