

Restoring backup



- [Restore folders](#)
 - [Restore database](#)
 - [MariaDB](#)
 - [MySQL](#)
 - [PostgreSQL](#)
 - [Restoring](#)
 - [MariaDB](#)
 - [MySQL](#)
 - [SQLite](#)
 - [PostgreSQL](#)
 - [Synchronising with clients after data recovery](#)
-

To restore a Nextcloud installation there are four main things you need to restore:

1. The configuration directory
2. The data directory
3. The database
4. The theme directory



You must have the database, data directory, and configuration files. You cannot complete restoration without all three.

Restore folders



This guide assumes that your previous backup is called "nextcloud-dirbkp"

Simply copy your configuration and data folder (or even your whole Nextcloud install and data folder) to your Nextcloud environment. You could use this command:

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

Restore database

“ ”

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.

MariaDB

MariaDB is the recommended database engine. To restore MariaDB using the [mariadb](#) client:

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

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

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

MySQL

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

“ ”

This guide assumes that your previous backup is called "nextcloud-sqlbkp.bak"

MariaDB

MariaDB is the recommended database engine. To restore MariaDB using the [mariadb](#) client:

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

MySQL

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
```

Synchronising with clients after data recovery

By default the Nextcloud server is considered the authoritative source for the data. If the data on the server and the client differs clients will default to fetching the data from the server.

If the recovered backup is outdated the state of the clients may be more up to date than the state of the server. In this case also make sure to run the [maintenance:data-fingerprint](#) command afterwards. It changes the logic of the synchronisation algorithm to try and recover as much data as possible. Files missing on the server are therefore recovered from the clients and in case of different content the users will be asked.

This can also help in rare scenarios when the database is newer than the data directory. The server will restore the data from the clients and preserve the shares. Until then the files would be visible but not accessible. A [files:scan](#) is required afterwards to update the database.

“ ”

The usage of `maintenance:data-fingerprint` can cause conflict dialogues and difficulties deleting files on the client. Therefore it's only recommended to prevent dataloss if the backup was outdated. This command does not require the server to be in maintenance mode.

If you are running multiple application servers you will need to make sure the config files are synced between them so that the updated data-fingerprint is applied on all instances.

#3

□ cocobook-sync ### 2026-10-28 16:30:38 UTC

□ cocobook-sync ### 2026-10-29 01:09:26 UTC