

SMB/CIFS



- [Case sensitive file system](#)
 - [SMB update notifications](#)
 - [SMB authentication](#)
 - [Decrease sync delay](#)
 - [Hidden files upload failure or not shown](#)
-

Nextcloud can connect to Windows file servers or other SMB-compatible servers with the SMB/CIFS backend.



The SMB/CIFS backend requires `smbclient` or the PHP smbclient module (`php-smbclient` / `libsmbclient-php`) to be installed on the Nextcloud server. The PHP smbclient module is **strongly preferred** — without it, the backend falls back to spawning the `smbclient` binary, which has known limitations (notably, downloads of files larger than ~512 MB from external SMB shares can fail; see [nextcloud/server#31308](#)). These should be included in any Linux distribution.

(See [PECL smbclient](#) if your distro does not include them.)

You need the following information:

- Folder name for your local mountpoint.
- Host: The URL of the Samba server.
- Username: The username or `domain\username` (see below) used to login to the Samba server.
- Password: the password to login to the Samba server.
- Share: The share on the Samba server to mount.
- Remote Subfolder: The remote subfolder inside the Samba share to mount (optional, defaults to /). To assign the Nextcloud logon username automatically to the subfolder, use `$user` instead of a particular subfolder name.
- And finally, the Nextcloud users and groups who get access to the share.

Optionally, you can specify a `Domain`. This is useful in cases where the SMB server requires a domain and a username, and an advanced authentication mechanism like session credentials is used so that the username cannot be modified. This is concatenated with the username, so the backend gets `domain\username`



For improved reliability and performance, we recommended installing `libsmbclient-php`, a native PHP module for connecting to SMB servers.

smbcifs SMB / CIFS

Session credentials
Username and password
Session credentials

smbserver
users
/shared
Domain

All users. Type to select

See [External Storage](#) for additional mount options and information.

See [External Storage authentication mechanisms](#) for more information on authentication schemes.

Case sensitive file system

This option tells Nextcloud whether the SMB share is backed by a case-sensitive filesystem. It does not change how the SMB server stores files, it only affects how Nextcloud performs lookups and validates paths.

- **Enabled (default):** Nextcloud assumes the share is case-sensitive. File existence checks use exact case and case-only renames are allowed.
- **Disabled:** Nextcloud assumes the share is case-insensitive and compensates by scanning directory entries to find the exact name used on the server. It also blocks renames that only change letter case. This can be slower on large directories.

If the setting does not match the SMB server's filesystem, you can run into confusing behavior:

- Case-insensitive SMB (for example NTFS or FAT) with the option enabled can lead to cache inconsistencies or conflicts, because Nextcloud may treat `File.txt` and `file.txt` as separate while the server treats them as the same file.
- Case-sensitive SMB (for example ext4) with the option disabled can cause unnecessary directory scans and case-only renames being rejected, even though the server itself would allow them.

SMB update notifications

“ ”

[Detecting changes made outside Nextcloud](#) explains why external changes to a mounted storage are not detected automatically and how update notifications fit into that picture.

Nextcloud can use smb update notifications to listen for changes made to a configured SMB/CIFS storage and detect external changes made to the storage in near real-time.

“ ”

Due to limitations of linux based SMB servers, this feature only works reliably on Windows SMB servers.

“ ”

Using update notifications requires `smbclient` 4.x or newer. Due to limitations with the `smbclient` PHP module, the `smbclient` binary is required even when using the PHP module.

To start listening to update notifications, start the `occ` command like this:

```
occ files_external:notify <mount_id>
```

You can find the mount id for a specific storage using `occ files_external:list`

On default this command shows no output, can you see the list of detected changes by passing the `-v` option to the command.

SMB authentication

Update notifications are not supported when using 'Login credentials, save in session' authentication. Using update notifications is only supported with 'Login credentials, save in database'.

Even when using 'Login credentials, save in database' or 'User entered, stored in database' authentication the notify process can not use the credentials saved to attach to the smb shares because the notify process does not run in the context of a specific user in those cases you can provide the username and password using the `--username` and `--password` arguments.

Decrease sync delay

Any updates detected by the notify command will only be synced to the client after the Nextcloud cron job has been executed (usually every 15 minutes). If this interval is too high for your use case, you can decrease it by running `occ files:scan --unscanned --all` at the desired interval. Note that this might increase the server load and you'll need to ensure that there is no overlap between runs.

Hidden files upload failure or not shown

If you have the configuration `hide dot files = Yes`, you will not be able to upload a hidden file (dot file) nor will you be able to show hidden files on your filelist (even if the 'show hidden file' option is checked on the nextcloud settings. Make sure you have the following option in your configuration:

```
hide dot files = No
```

📄 #3

📄 cocobook-sync 📄 2026-10-28 16:27:58 UTC

📄 cocobook-sync 📄 2026-10-29 01:09:19 UTC