

Nextcloud-Ordner als Live-Mount auf Ubuntu Server (rclone)

Diese Anleitung richtet einen ausgewählten Nextcloud-Ordner (z. B. `AI`) als echtes, dauerhaft eingebundenes Dateisystem auf einem Ubuntu-Server ein – per WebDAV über `rclone mount`. Änderungen werden nahezu in Echtzeit zwischen Server und Nextcloud abgeglichen (kein Polling-Sync, kein Desktop-Client, kein Xvfb nötig).

Einsatzzweck: Verteiltes Gedächtnis / Datenordner für Anwendungen (z. B. KI-Agenten), die kontinuierlich lesend und schreibend auf einen Nextcloud-Ordner zugreifen sollen, auf Servern ohne Desktop-Umgebung.

Voraussetzungen

- Ubuntu Server mit SSH-Zugriff
 - Ein Nextcloud-Account mit App-Passwort (siehe Schritt 1)
 - `rclone` und `fuse3`
-

Schritt 1: App-Passwort in Nextcloud erstellen

1. Nextcloud im Browser öffnen und normal einloggen (Login + ggf. OTP).
 2. Profilbild oben rechts → **Persönliche Einstellungen**.
 3. Reiter **Sicherheit** (links).
 4. Ganz nach unten scrollen zu **App-Passwörter**.
 5. Namen vergeben (z. B. `Ubuntu-Server-AI-Mount`) → **Neues App-Passwort erstellen**.
 6. Generiertes Passwort kopieren – wird in Schritt 3 benötigt.
-

Schritt 2: Pakete installieren

```
sudo apt update  
sudo apt install -y rclone fuse3
```

“Kein `nextcloud-desktop`, kein `xvfb` nötig – rclone kommt ohne GUI-Abhängigkeiten aus.

Schritt 3: rclone-Remote konfigurieren

Interaktive Einrichtung:

```
rclone config
```

- `n` (New remote)
- Name: `Nextcloud` (Groß-/Kleinschreibung beachten – wird später exakt so referenziert, inkl. Doppelpunkt: `Nextcloud:`)
- Typ: `webdav`
- URL:

```
https://DEINE-NEXTCLOUD-DOMAIN/remote.php/dav/files/DEIN_USERNAME/
```

- Vendor: `nextcloud`
- User: dein Nextcloud-Login
- Password: das App-Passwort aus Schritt 1 (rclone verschlüsselt es automatisch in der Config)
- Rest mit Enter bestätigen, am Ende mit `y` speichern

Alternative - nicht-interaktiv:

```
rclone config create Nextcloud webdav \  
url=https://DEINE-NEXTCLOUD-DOMAIN/remote.php/dav/files/DEIN_USERNAME/ \  
vendor=nextcloud \  
user=DEIN_USERNAME \  
pass=DEIN_APP_PASSWORT
```

Verbindung testen:

```
rclone ls Nextcloud:
```

Sollte die Ordnerliste deiner Nextcloud anzeigen (Doppelpunkt nicht vergessen – ohne `:` interpretiert rclone den Namen als lokalen Pfad statt als Remote!).

Schritt 4: Mount-Punkt anlegen

```
mkdir -p "/pfad/zu/deinem/mount-ordner"
```

Beispiel für den `AI`-Ordner:

```
mkdir -p "/home/DEIN_USER/nextcloud-ai"
```

Schritt 5: Manuell testen

```
rclone mount Nextcloud:AI "/home/DEIN_USER/nextcloud-ai/" \  
--vfs-cache-mode writes \  
--daemon
```

- `Nextcloud:AI` → nur dieser Unterordner wird gemountet (beliebiger Pfad möglich, auch verschachtelt: `Nextcloud:AI/Unterordner`, oder `Nextcloud:` für den kompletten Account)
- `--daemon` → läuft im Hintergrund, Terminal bleibt frei (nur für manuelle Tests; im systemd-Service **ohne** `--daemon`, siehe Schritt 6)

Prüfen:

```
ls -la "/home/DEIN_USER/nextcloud-ai/"
```

Test-Datei anlegen und in der Nextcloud-Weboberfläche prüfen, ob sie ankommt (kurze Verzögerung durch VFS-Cache ist normal):

```
echo "test" > "/home/DEIN_USER/nextcloud-ai/server-test.txt"
```

Zum Beenden des Test-Mounts:

```
fusermount -u "/home/DEIN_USER/nextcloud-ai/"
```

Schritt 6: Als systemd-User-Dienst dauerhaft einrichten

Damit der Mount Server-Neustarts übersteht und automatisch wieder da ist.

1. Service-Datei anlegen:

```
mkdir -p ~/.config/systemd/user  
nano ~/.config/systemd/user/nextcloud-ai-mount.service
```

2. Inhalt einfügen (Pfade und Remote-Ordner anpassen):

```
[Unit]  
Description=Nextcloud AI-Ordner als WebDAV-Mount  
After=network-online.target  
Wants=network-online.target  
  
[Service]  
Type=notify  
ExecStart=/usr/bin/rclone mount Nextcloud:AI "/home/DEIN_USER/nextcloud-ai/" --vfs-cache-mode writes --vfs-cache-max-age 24h  
ExecStop=/bin/fusermount -u "/home/DEIN_USER/nextcloud-ai/"  
Restart=always  
RestartSec=10  
  
[Install]  
WantedBy=default.target
```

“Kein `--daemon`-Flag hier: Der Prozess muss im Vordergrund bleiben, damit systemd ihn korrekt als laufenden Dienst überwachen kann. Dass der Befehl "blockiert" (nicht ins Terminal zurückkehrt), ist bei `rclone mount` ohne `--daemon` normales, gewolltes Verhalten.

3. Aktivieren:

```
systemctl --user daemon-reload
systemctl --user enable --now nextcloud-ai-mount
```

4. Sicherstellen, dass der Dienst auch ohne aktive SSH-Session weiterläuft (Linger – einmalig nötig):

```
sudo loginctl enable-linger $USER
```

Prüfen:

```
loginctl show-user $USER | grep Linger
```

Sollte `Linger=yes` zeigen.

Schritt 7: Status & Logs prüfen

```
systemctl --user status nextcloud-ai-mount
ls -la "/home/DEIN_USER/nextcloud-ai/"
```

Live-Logs:

```
journalctl --user -u nextcloud-ai-mount -f
```

“ Falls `journalctl --user` ohne Ergebnisse bleibt oder Berechtigungsfehler zeigt: `sudo journalctl --user-unit=nextcloud-ai-mount -f` verwenden, oder sich einmal komplett neu per SSH einloggen, nachdem man mit `sudo usermod -aG systemd-journal $USER` zur Gruppe hinzugefügt wurde.

Wichtige Hinweise

- **--vfs-cache-mode writes**: Schreibzugriffe werden lokal gepuffert und zuverlässig hochgeladen, auch bei kurzen Netzausfällen.
- **Kein Polling-Sync**: Änderungen gehen nahezu in Echtzeit über WebDAV, nicht erst beim nächsten Sync-Zyklus wie bei `rclone sync` / `bisync`.

- **rclone bisync** **bewusst vermieden**: Ist offiziell als Beta-Feature eingestuft – für ein produktiv genutztes, kontinuierlich beschriebenes Verzeichnis ist `mount` die robustere Wahl.
- **Mehrere Ordner**: Für weitere Nextcloud-Ordner einfach ein zweites Service-Paar mit anderem Namen (z. B. `nextcloud-technik-mount.service`) und anderem `targetPath` (`Nextcloud:Technik`) anlegen – läuft unabhängig parallel.
- **Remote-Name ist case-sensitive**: `Nextcloud:` $\neq$ `nextcloud:`.

Troubleshooting-Kurzreferenz

Symptom	Ursache	Lösung
<code>rclone ls</code> zeigt nichts	Remote-Name falsch geschrieben / Doppelpunkt vergessen	<code>rclone listremotes</code> prüfen, exakte Schreibweise verwenden
<code>didn't find section in config file</code>	Remote-Name in Config weicht ab	<code>rclone config show</code> bzw. <code>cat ~/.config/rclone/rclone.conf</code> prüfen
Mount-Befehl "hängt" im Terminal	Normales Verhalten ohne <code>--daemon</code>	Für Tests <code>--daemon</code> nutzen, im systemd-Service bewusst weglassen
<code>journalctl --user</code> verweigert Zugriff	Fehlende Gruppenmitgliedschaft <code>systemd-journal</code>	<code>sudo usermod -aG systemd-journal \$USER</code> , neu einloggen