

Servizi self-hosted

- [Nextcloud](#)
- [Immich](#)

Nextcloud

Comandi Docker per Nextcloud (occ)

“ **Premessa** I comandi `occ` vanno **sempre** eseguiti come utente `www-data`. Sostituisci `CONTAINER` con il nome del tuo container Nextcloud (lo trovi con `docker ps`).

occ — comando base

```
docker exec -u www-data CONTAINER php occ <comando>
docker exec -u www-data CONTAINER php occ list    # elenco di TUTTI i comandi
```

Manutenzione

```
# Modalità manutenzione (attiva prima di interventi pesanti / backup)
docker exec -u www-data CONTAINER php occ maintenance:mode --on
docker exec -u www-data CONTAINER php occ maintenance:mode --off

# Riparazioni dopo aggiornamenti o problemi
docker exec -u www-data CONTAINER php occ maintenance:repair

# Stato generale dell'installazione
docker exec -u www-data CONTAINER php occ status
```

File (ripristino da snapshot / backup)

Quando si ripristinano file direttamente sul filesystem, Nextcloud non li "vede" finché non si rilancia la scansione che risincronizza il database con il disco.

```
docker exec -u www-data CONTAINER php occ files:scan --all
docker exec -u www-data CONTAINER php occ files:scan UTENTE
docker exec -u www-data CONTAINER php occ files:scan --path="/UTENTE/files/Cartella"
docker exec -u www-data CONTAINER php occ files:cleanup # rimuove voci orfane dal DB
```

“ **Nota:** il `--path` parte sempre da `/utente/files/...`, non dal percorso assoluto sul disco. I file ripristinati devono trovarsi nel volume dati montato nel container (di solito `/var/www/html/data`) e appartenere a `www-data`.

Correggere il proprietario dei file ripristinati

```
docker exec -u www-data CONTAINER find /var/www/html/data/UTENTE/files/Cartella -exec chown www-data:www-data {} \;
```

Utenti

```
docker exec -u www-data CONTAINER php occ user:list
docker exec -u www-data CONTAINER php occ user:add NOMEUTENTE
docker exec -u www-data CONTAINER php occ user:resetpassword NOMEUTENTE
docker exec -u www-data CONTAINER php occ user:setting NOMEUTENTE files quota 10GB
```

App

```
docker exec -u www-data CONTAINER php occ app:list
docker exec -u www-data CONTAINER php occ app:enable NOMEAPP
docker exec -u www-data CONTAINER php occ app:disable NOMEAPP
docker exec -u www-data CONTAINER php occ app:update --all
```

Configurazione e indici database

```
# Leggi/scrivi parametri di config.php
docker exec -u www-data CONTAINER php occ config:list system
docker exec -u www-data CONTAINER php occ config:system:set key --value="valore"

# Indici e colonne mancanti (warning tipici nella pagina "Panoramica")
docker exec -u www-data CONTAINER php occ db:add-missing-indices
docker exec -u www-data CONTAINER php occ db:add-missing-columns
docker exec -u www-data CONTAINER php occ db:convert-filecache-bigint
```

Cron / anteprime

```
docker exec -u www-data CONTAINER php occ background:cron      # imposta cron come metodo
job
docker exec -u www-data CONTAINER php occ preview:generate-all # rigenera anteprime
```

Suggerimento: alias per scrivere meno

Crea un alias nella shell dell'host per non riscrivere ogni volta la parte lunga:

```
alias nocc='docker exec -u www-data CONTAINER php occ'
```

Poi basta scrivere, ad esempio:

```
nocc files:scan --all
nocc status
```

Se usi **docker compose**, l'equivalente del comando base è:

```
docker compose exec -u www-data app php occ <comando>
```

(sostituisci `app` con il nome del servizio Nextcloud nel tuo `docker-compose.yml`).

Immich

Comandi Docker per Immich

“ I container principali di Immich sono `immich_server`, `immich_machine_learning` e `immich_postgres`. Tutti i comandi si eseguono dalla cartella dove si trova il `docker-compose.yml`.

immich-admin — CLI amministrativa

Il container `immich_server` include una CLI integrata chiamata `immich-admin`. Prima connettiti al container, poi esegui i comandi:

```
docker exec -it immich_server bash  
  
# una volta dentro:  
immich-admin help
```

Oppure direttamente senza entrare nel container:

```
docker exec -it immich_server immich-admin <comando>
```

Comando	Descrizione
<code>help</code>	Mostra tutti i comandi disponibili
<code>version</code>	Versione installata di Immich
<code>list-users</code>	Elenca gli utenti
<code>reset-admin-password</code>	Resetta la password dell'admin
<code>enable-password-login</code>	Abilita il login con password
<code>disable-password-login</code>	Disabilita il login con password

Comando	Descrizione
<code>enable-oauth-login</code>	Abilita il login OAuth
<code>disable-oauth-login</code>	Disabilita il login OAuth
<code>enable-maintenance-mode</code>	Attiva la modalità manutenzione
<code>disable-maintenance-mode</code>	Disattiva la modalità manutenzione
<code>change-media-location</code>	Aggiorna i path nel DB dopo aver spostato i media

Esempi pratici

```
# Resettare la password dell'admin
docker exec -it immich_server immich-admin reset-admin-password

# Attivare la modalità manutenzione
docker exec -it immich_server immich-admin enable-maintenance-mode
# → restituisce un URL con token per accedere alla manutenzione

# Disattivarla dopo l'intervento
docker exec -it immich_server immich-admin disable-maintenance-mode

# Vedere la versione
docker exec -it immich_server immich-admin version
```

Backup del database

Il backup automatico è attivo per default (ogni giorno alle 2:00, ultimi 14 backup). I file vengono salvati in `UPLOAD_LOCATION/backups`.

Per avviare un backup manuale via web UI: **Administration** → **Job Queues** → **Create job** → **Create Database Backup**

Per fare un backup manuale da riga di comando:

```
docker exec -t immich_postgres pg_dumpall \
  --clean --if-exists \
  --username=<DB_USERNAME> \
  > /percorso/backup/immich-database.sql
```

Con compressione:

```
docker exec -t immich_postgres pg_dumpall \  
  --clean --if-exists \  
  --username=<DB_USERNAME> \  
  | gzip > /percorso/backup/immich-database.sql.gz
```

“ **Attenzione:** il backup del database contiene solo i metadati, non le foto/video. Per un backup completo occorre copiare anche la cartella `UPLOAD_LOCATION`.

Restore del database

Il metodo raccomandato è via web UI: **Administration** → **Maintenance** → **Restore database backup**

Per il restore da riga di comando (richiede un'installazione fresca dove il server non ha mai girato):

```
# 1. Crea i container senza avviarli  
docker compose create  
  
# 2. Avvia solo Postgres  
docker start immich_postgres  
sleep 10  
  
# 3. Esegui il restore  
docker exec -i immich_postgres psql \  
  --dbname=postgres \  
  --username=<DB_USERNAME> \  
  < /percorso/backup/immich-database.sql  
  
# 4. Avvia tutti gli altri container  
docker compose up -d
```

“ Se il server è già stato avviato prima del restore, elimina la cartella `DB_DATA_LOCATION` per azzerare il database prima di procedere.

Permessi e proprietario dei file

Se hai ripristinato file direttamente sul filesystem, controlla che il proprietario sia corretto:

```
# Trova l'UID usato da Immich (di default 1000)
docker exec immich_server id

# Correggi il proprietario nella cartella upload
sudo chown -R 1000:1000 /percorso/UPLOAD_LOCATION
```