

NSC - Charis - ITA

Guida completa alla configurazione, installazione e migrazione

Indice

1. [WebDAV / Nextcloud](#)
2. [Google Drive](#)
3. [Tipologie e Gestione Dati](#)
4. [Email / SMTP](#)
5. [Guida all'Installazione](#)
6. [Backup e Migrazione Server](#)
7. [Docker](#)

1. WebDAV / Nextcloud

Cos'è

WebDAV è un protocollo che permette di accedere a file su server remoti. Nextcloud lo utilizza per la sincronizzazione dei file.

Configurazione

Vai in **Impostazioni** → **Repository** e crea un nuovo repository di tipo WebDAV.

Campi obbligatori

Nome

Un nome descrittivo (es. "Nextcloud Lavoro")

Base URI

L'URL del tuo server WebDAV.

Nextcloud: `https://tuodominio.it/remote.php/dav/files/utente/`

Altri: `https://server.webdav.it/`

Root

Percorso base all'interno del server (es. `/Documenti/`). Lascia `/` per usare la root.

Username

Il tuo nome utente

Password

La password dell'account

Auth Type

`Basic` per la maggior parte dei server (Nextcloud, ownCloud).

Nome utente con @ (chiocciola)

Se il tuo nome utente contiene il carattere @ (es. `mario.rossi@azienda.com`), segui queste regole:

- **Nel campo "Base URI":** l'indirizzo email deve essere **URL-encoded**:

`https://server.it/remote.php/dav/files/mario.rossi%40azienda.com/`

(il carattere @ diventa %40)

- **Nel campo "Username":** inserisci l'indirizzo email **normale** con la @ vera.

Il sistema gestisce automaticamente la differenza tra l'URI (che richiede %40) e l'autenticazione (che usa la @ reale).

Test connessione

Dopo aver salvato, usa il pulsante **Test** nella tabella dei repository per verificare che la connessione funzioni.

Importare file

Apri la sezione **Documenti**, clicca sul pulsante del repository nella toolbar per sfogliare i file remoti, poi clicca **Salva tra i documenti** su un file per importarlo come documento locale.

2. Google Drive

Cos'è

Integrazione con Google Drive per sfogliare e importare file direttamente dal tuo spazio Google Drive.

Requisiti

1. Un account Google / Google Workspace
2. Un progetto sulla **Google Cloud Console** con l'API Google Drive abilitata
3. Credenziali OAuth 2.0 (Client ID e Client Secret)

Passo 1: Creare il progetto Google Cloud

1. Vai su **Google Cloud Console** (console.developers.google.com)
2. Crea un nuovo progetto o selezionane uno esistente
3. Vai a **API e servizi** → **Libreria**
4. Cerca "**Google Drive API**" e **abilitala**
5. Vai a **API e servizi** → **Credenziali**
6. Clicca "**Crea credenziali** → **ID client OAuth**"
7. Tipo applicazione: "**Applicazione Web**"
8. **URI di reindirizzamento autorizzato**: aggiungi <http://localhost/auth/google-drive/callback>
9. Completa la creazione e **copia Client ID e Client Secret**

Passo 2: Configurazione nell'app

1. Vai in **Impostazioni** → **Repository**
2. Clicca "**Nuovo Repository**" e seleziona tipo **Google Drive**
3. Inserisci un nome descrittivo (es. "Drive Lavoro")
4. Incolla **Client ID** e **Client Secret**
5. **Folder ID (opzionale)**: se vuoi limitare l'accesso a una cartella specifica, inserisci il suo ID (lo trovi nell'URL quando apri la cartella in Google Drive: https://drive.google.com/drive/folders/ID_QUI)
6. Clicca "**Autorizza Google Drive**" — verrai reindirizzato a Google per concedere i permessi
7. Dopo l'autorizzazione, il refresh token verrà salvato automaticamente

Google Drive API non abilitata

Se vedi l'errore "*Google Drive API non abilitata*" nel test connessione, significa che l'API Drive non è stata attivata nel tuo progetto Google Cloud.

Soluzione

Vai su <https://console.developers.google.com/apis/api/drive.googleapis.com/overview>

Seleziona il progetto corretto dall'header in alto, clicca **"Enable"** e attendi qualche minuto prima di riprovare.

Importare file

Stessa procedura di WebDAV: apri **Documenti**, clicca sul repository Drive, sfoglia i file e usa **Salva tra i documenti**.

File Google (Documenti, Fogli, Presentazioni)

I file creati con Google Workspace (Documenti Google, Fogli Google, Presentazioni Google, Disegni Google) non possono essere scaricati direttamente perché non hanno un formato binario nativo.

Il sistema li converte automaticamente in formato Office Open XML durante l'import:

File Google	Formato convertito
Documento Google	.docx (Word)
Foglio Google	.xlsx (Excel)
Presentazione Google	.pptx (PowerPoint)
Disegno Google	.png (Immagine)
Altri (Moduli, Script)	.pdf (PDF)

Il nome del file viene automaticamente esteso con l'estensione corretta (es. "Relazione" → "Relazione.docx").

3. Tipologie e Gestione Dati

Tipologie Documenti

Le tipologie classificano i documenti (avatar, galleria, documento, statuto, altro).

Si gestiscono in **Impostazioni** → **Tipologie Documenti**. Puoi aggiungere, rinominare, riordinare ed eliminare tipologie (tranne quelle con documenti associati).

Tipologie Individui

Classificano gli individui (es. socio, volontario, dipendente). Si gestiscono in **Impostazioni** → **Tipologie**.

Tipologie Eventi

Classificano gli eventi (es. riunione, assemblea, formazione). Si gestiscono in **Impostazioni** → **Tipologie Eventi**.

Ruoli Individui

Definiscono il ruolo o la carica di un individuo (es. presidente, segretario, tesoriere). Si gestiscono in **Impostazioni** → **Ruoli**.

Visibilità Documenti

Ogni documento può avere diversi livelli di visibilità:

- **Pubblico** — visibile a tutti gli utenti
- **Individuo** — visibile solo a uno specifico individuo
- **Gruppo** — visibile solo ai membri di un gruppo
- **Evento** — visibile solo ai partecipanti di un evento
- **Mailing** — visibile solo ai membri di una mailing list

4. Email / SMTP

Configurazione Email

Vai in **Impostazioni** → **Email** per configurare la posta elettronica.

Server IMAP (Ricezione)

Usato per ricevere e leggere le email direttamente nell'app.

Campi consigliati per provider comuni

Provider	Host	Porta	Crittografia
Gmail	imap.gmail.com	993	SSL
Outlook/Office 365	outlook.office365.com	993	SSL
Libero	imapmail.libero.it	993	SSL
Aruba	imap.aruba.it	993	SSL

Server SMTP (Invio)

Usato per inviare email dall'app. Supporta account multipli (mittenti).

Campi consigliati per provider comuni

Provider	Host	Porta	Crittografia
Gmail	smtp.gmail.com	587	TLS
Outlook/Office 365	smtp.office365.com	587	TLS
Libero	smtp.libero.it	465	SSL
Aruba	smtp.aruba.it	465	SSL

Gmail / Google Workspace

Gmail richiede una **Password per l'app** (non la password normale) se usi la verifica in due passaggi. Puoi generarla su <https://myaccount.google.com/apppasswords>.

Mailing

Le **Mailing List** permettono di inviare email a gruppi predefiniti di destinatari.

- Crea una mailing list in **Mailing List** → **Nuova**
- Puoi popolare la lista da individui, gruppi o manualmente
- Invia email alla lista da **Email** → **Nuova Email**, selezionando la mailing list come destinatario

Test connessione

Dopo aver configurato i server, usa il pulsante **Test Connessione** per verificare che IMAP e SMTP funzionino correttamente.

5. Guida all'Installazione

Panoramica

L'applicazione include un **wizard di installazione web** che guida passo-passo nella configurazione su un nuovo server LAMP. Supporta due modalità:

Modalità	Descrizione
Fresh Install	Installa l'applicazione da zero su un nuovo server: crea il database, esegue le migration, popola i dati di base e crea l'amministratore.
Restore da Backup	Ripristina un'installazione esistente da un file ZIP di backup: importa il database, ripristina i documenti e la configurazione.

Requisiti del Server

Componente	Versione Minima	Note
Web Server	Apache 2.4+	Con mod_rewrite abilitato
PHP	8.2+ (consigliato 8.4)	Estensioni: pdo_mysql, mbstring, xml, curl, zip, gd, fileinfo
MySQL / MariaDB	MySQL 8.0+ / MariaDB 10.5+	Supporto utf8mb4
Composer	2.x	Per gestione dipendenze PHP
Node.js / npm	Node 20+ / npm 10+	Per compilazione asset frontend (opzionale se precompilati)

Installazione Passo-Passo

Passo 1: Preparare il server LAMP

```
sudo apt update && sudo apt upgrade -y

sudo apt install -y apache2 mariadb-server php8.4 php8.4-cli \
  php8.4-mysql php8.4-xml php8.4-mbstring php8.4-curl \
  php8.4-zip php8.4-bcmath php8.4-gd php8.4-intl \
  composer git unzip

sudo a2enmod rewrite
sudo systemctl restart apache2
```

Passo 2: Ottenere l'applicazione

Puoi clonare il repository con git oppure estrarre il file ZIP ricevuto nella cartella `/var/www/glastree`.


```
cd /var/www

# Opzione A — Git clone
git clone <URL_REPOSITORY> glastree

# Opzione B — ZIP (se non hai git)
# unzip /percorso/del/glastree.zip -d glastree

cd glastree
composer install --no-dev --optimize-autoloader
npm ci && npm run build
```

Se non hai accesso a npm, puoi saltare il build. L'interfaccia userà comunque AdminLTE via CDN.

Passo 3: Permessi cartelle

```
sudo chown -R www-data:www-data storage bootstrap/cache
sudo chmod -R 775 storage bootstrap/cache
```

Passo 4: Configurare Apache

```
sudo tee /etc/apache2/sites-available/glastree.conf <<APACHE
<VirtualHost *:80>
    ServerName glastree.esempio.it
    DocumentRoot /var/www/glastree/public

    <Directory /var/www/glastree/public>
        Options -Indexes +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog \${APACHE_LOG_DIR}/glastree_error.log
    CustomLog \${APACHE_LOG_DIR}/glastree_access.log combined
</VirtualHost>
APACHE

sudo a2dissite 000-default.conf
sudo a2ensite glastree.conf
sudo systemctl reload apache2
```

Passo 5: Avviare il wizard di installazione

Apri il browser e visita: <http://glastree.esempio.it/installer/>

Il wizard ti guiderà attraverso 6 passi:

Passo	Descrizione
1. Benvenuto	Scegli Fresh Install o Restore da Backup
2. Requisiti	Verifica PHP, estensioni e permessi
3. Database	Configura MySQL (crea nuovo o usa esistente)
4. Installazione	Fresh: migration e seed. Backup: upload ZIP
5. Amministratore	Fresh: crea super-admin. Backup: verifica utenti
6. Finalizzazione	Cache, storage link, auto-eliminazione installer

Attenzione: Al termine dell'installazione, la cartella `installer/` viene **automaticamente eliminata**. Non puoi rieseguire l'installazione se non ricreando manualmente la cartella.

Modalità Fresh Install

Seleziona questa modalità quando installi l'applicazione per la prima volta su un server.

Cosa succede durante l'installazione:

1. Genera `APP_KEY` per la crittografia
2. Esegue tutte le migration per creare le tabelle del database
3. Popola i dati di base (diocesi, comuni, ruoli, preset)
4. Crea il collegamento `storage → public/storage`
5. Crea l'utente amministratore con i permessi completi

Modalità Restore da Backup

Seleziona questa modalità quando vuoi migrare un'installazione esistente su un nuovo server.

Cosa serve:

- Un file ZIP generato dalla pagina **Admin** → **Backup** del server originale
- Il wizard accetta upload diretto del file o un percorso sul server

Cosa succede durante il restore:

1. Estrae il file ZIP in una cartella temporanea
2. Importa `database.sql` nel database MySQL configurato
3. Copia i file documenti in `storage/app/documenti/`
4. Ripristina il file `.env` dal backup (preservando le credenziali DB del nuovo server)

Dopo l'Installazione

1. **Configura SSL** con Let's Encrypt:

```
sudo apt install -y certbot python3-certbot-apache
sudo certbot --apache -d glastree.esempio.it
```

2. **Accedi** con le credenziali amministratore create durante l'installazione

3. **Configura l'applicazione** in Impostazioni:

- Nome applicazione e logo
- Email (SMTP per invio, IMAP per ricezione)
- Repository remoti (Google Drive, Nextcloud)

4. **Crea utenti** dal pannello Admin → Utenti

5. **Configura il cron** per backup automatico e sync email:

```
sudo crontab -u www-data -e
# Inserisci:
* * * * * cd /var/www/glastree && php artisan schedule:run >> /dev/null 2>&1
```

Risoluzione Problemi

Errore "500 | Server Error" dopo l'installazione

Probabilmente mancano permessi sulle cartelle `storage` o `bootstrap/cache` :

```
sudo chown -R www-data:www-data storage bootstrap/cache
sudo chmod -R 775 storage bootstrap/cache
```

Errore "No application encryption key"

APP_KEY non generata. Nella cartella dell'app:

```
php artisan key:generate
```

mod_rewrite non funziona

Se le rotte restituiscono 404, verifica che Apache abbia AllowOverride:

```
sudo a2enmod rewrite
# Nel VirtualHost: AllowOverride All
sudo systemctl reload apache2
```

6. Backup e Migrazione Server

Cos'è

Il sistema di backup crea un archivio ZIP contenente tutto il necessario per ripristinare l'applicazione su un nuovo server. Include database, file caricati e configurazione.

Cosa include il backup

Componente	Incluso	Dettagli
Database (MySQL)	Sempre	Struttura, dati, stored procedure, eventi, trigger
File caricati	Opzionale	Documenti, avatar, gallerie
Configurazione .env	Opzionale	APP_KEY, credenziali DB, SMTP, token OAuth
Metadati	Sempre	Versione app, PHP, Laravel

Come eseguire un backup

Via Interfaccia Web	Via Comando
1. Vai su Admin → Backup 2. Verifica le impostazioni (includi files/.env) 3. Clicca Esegui Backup Ora 4. Scarica il file ZIP generato	<pre>php artisan backup:run</pre> Il file viene creato in <code>storage/app/backups/</code>

Backup automatico

Puoi abilitare il backup automatico dalla pagina **Admin** → **Backup**:

- **Frequenza:** giornaliero, settimanale o mensile
- **Ora:** consigliata 03:00 (notte)
- **Requisito:** il cron di Laravel deve essere configurato sul server

Per il backup automatico, aggiungi questo cron:

```
* * * * * cd /var/www/html/glastree && php artisan schedule:run >> /dev/null 2>&1
```

Procedura di Migrazione Completa

Segui questa procedura per migrare l'applicazione su un nuovo server LAMP (Linux, Apache, MySQL, PHP).

ATTENZIONE: APP_KEY

L'APP_KEY cripta i dati sensibili (password email SMTP, token OAuth Google Drive, ecc.).

- **Se hai il .env originale dal backup:** NON rigenerare APP_KEY — mantieni i dati criptati accessibili.
- **Se hai perso il .env:** DEVI rigenerare APP_KEY, ma perderai l'accesso ai dati criptati esistenti. Dovrai riconfigurare SMTP e OAuth.

Passo 1: Preparazione del server

```
sudo apt update && sudo apt upgrade -y
```

Passo 2: Installa LAMP + PHP 8.4

```
sudo apt install -y apache2 mariadb-server php8.4 php8.4-cli \  
php8.4-mysql php8.4-xml php8.4-mbstring php8.4-curl \  
php8.4-zip php8.4-bcmath php8.4-gd php8.4-intl \  
composer git unzip  
  
sudo a2enmod rewrite
```

Passo 3: Ottieni l'applicazione

Puoi clonare il repository con git oppure estrarre il file ZIP ricevuto nella cartella `/var/www/glastree`.

```
cd /var/www  
  
# Opzione A – Git clone  
git clone <URL_REPOSITORY> glastree  
  
# Opzione B – ZIP (se non hai git)  
# unzip /percorso/del/glastree.zip -d glastree  
  
cd glastree  
composer install --no-dev --optimize-autoloader  
npm ci && npm run build
```

Passo 4: Carica il backup

Copia il file `backup.zip` sul nuovo server e estrailo:

```
mkdir -p /tmp/restore  
unzip backup.zip -d /tmp/restore/  
ls -la /tmp/restore/  
# Dovresti vedere: database.sql .env files/ manifest.json
```

Passo 5: Crea database e importa

```
sudo mysql -u root <<EOF  
CREATE DATABASE glastree CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
CREATE USER 'glastree'@'localhost' IDENTIFIED BY 'ScegliPasswordFortep4ss!';  
GRANT ALL ON glastree.* TO 'glastree'@'localhost';  
FLUSH PRIVILEGES;  
EOF  
  
mysql -u glastree -p glastree < /tmp/restore/database.sql
```

Passo 6: Configura .env

```
cp /tmp/restore/.env .env
# Modifica se necessario:
# APP_URL=https://nuovo-dominio.it
# DB_HOST=localhost
# DB_DATABASE=glastree
# DB_USERNAME=glastree
# DB_PASSWORD=ScegliPasswordFortep4ss!

# DECIDI SULL'APP_KEY (leggi avviso sopra!)
# php artisan key:generate ← SOLO se non hai il .env originale
```

Passo 7: Ripristina file e permessi

```
cp -r /tmp/restore/files/* storage/app/documenti/

sudo chown -R www-data:www-data storage bootstrap/cache
sudo chmod -R 755 storage bootstrap/cache

php artisan storage:link
```

Passo 8: Cache

```
php artisan optimize:clear
php artisan config:cache
php artisan route:cache
php artisan view:cache
```

Passo 9: Configura Apache

```
sudo tee /etc/apache2/sites-available/glastree.conf <<APACHE
<VirtualHost *:80>
    ServerName glastree.esempio.it
    DocumentRoot /var/www/glastree/public

    <Directory /var/www/glastree/public>
        Options -Indexes +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog \${APACHE_LOG_DIR}/glastree_error.log
    CustomLog \${APACHE_LOG_DIR}/glastree_access.log combined
</VirtualHost>
APACHE

sudo a2dissite 000-default.conf
sudo a2ensite glastree.conf
sudo systemctl reload apache2
```

Passo 10: SSL Let's Encrypt

```
sudo apt install -y certbot python3-certbot-apache
sudo certbot --apache -d glastree.esempio.it
```

Passo 11: Cron per backup automatico

```
sudo crontab -u www-data -e
# Inserisci:
* * * * * cd /var/www/glastree && php artisan schedule:run >> /dev/null 2>&1
```

Passo 12: Verifica finale

1. Apri <https://glastree.esempio.it> nel browser
 2. Fai login con le credenziali esistenti
 3. Verifica: Dashboard, Documenti, Impostazioni, Email
 4. Se hai rigenerato APP_KEY: riconfigura SMTP e OAuth Google Drive
-

PHP Web Installer

L'applicazione include un **wizard di installazione web** nella cartella `installer/`. Questo wizard gestisce sia installazioni fresh che ripristino da backup.

Vedi la sezione **Installazione** per la guida completa.

7. Docker

Creare un'immagine Docker con dati di base

Puoi creare un'immagine Docker pronta all'uso che include l'applicazione e i dati di base (tipologie documenti, tipologie eventi, ruoli). Scegli la configurazione in base alle tue esigenze:

Configurazione	Descrizione
SQLite (Leggero)	Database embedded, nessun servizio aggiuntivo. Ideale per demo, test, sviluppo locale o ambienti con poca concorrenza.
MySQL (Standard)	Database separato in un container MySQL. Configurazione standard, ideale per produzione, staging e ambienti con più utenti.

1. Dockerfile (comune a entrambe le modalità)

```
FROM php:8.4-apache

# Abilita mod_rewrite
RUN docker-php-ext-enable opcache \
    && a2enmod rewrite

# Installa estensioni PHP (sia SQLite che MySQL)
RUN docker-php-ext-install pdo_mysql pdo_sqlite bcmath gd zip

# Installa Composer
COPY --from=composer:2 /usr/bin/composer /usr/bin/composer

# Configura DocumentRoot per Laravel
ENV APACHE_DOCUMENT_ROOT=/var/www/html/public
RUN sed -ri -e 's!/var/www/html!${APACHE_DOCUMENT_ROOT}!g' /etc/apache2/sites-available/*.conf \
    && sed -ri -e 's!/var/www/!${APACHE_DOCUMENT_ROOT}!g' /etc/apache2/apache2.conf /etc/apache2/conf-
available/*.conf

# Copia l'applicazione
COPY . /var/www/html/

# Installa dipendenze e compila asset
RUN composer install --no-dev --optimize-autoloader \
    && mkdir -p storage/app/public \
    && mkdir -p storage/framework/cache/data \
    && mkdir -p storage/framework/sessions \
    && mkdir -p storage/framework/views \
    && mkdir -p storage/logs \
    && mkdir -p bootstrap/cache \
    && chown -R www-data:www-data storage bootstrap/cache

# Entrypoint per inizializzazione
COPY docker-entrypoint.sh /usr/local/bin/
RUN chmod +x /usr/local/bin/docker-entrypoint.sh

ENTRYPOINT ["docker-entrypoint.sh"]
CMD ["apache2-foreground"]
```


2. Entrypoint Script (comune)

Crea `docker-entrypoint.sh`. Lo script rileva automaticamente la configurazione in base alle variabili d'ambiente:

```
#!/bin/bash
set -e

# Se .env non esiste, lo genera dalle variabili d'ambiente
if [ ! -f .env ]; then
    cp .env.example .env

    if [ "$DB_CONNECTION" = "mysql" ]; then
        # Configura MySQL
        sed -i "s/DB_CONNECTION=mysql/DB_CONNECTION=mysql/" .env
        sed -i "s/DB_HOST=.*/DB_HOST=${DB_HOST:-mysql}/" .env
        sed -i "s/DB_PORT=.*/DB_PORT=${DB_PORT:-3306}/" .env
        sed -i "s/DB_DATABASE=.*/DB_DATABASE=${DB_DATABASE:-glastree}/" .env
        sed -i "s/DB_USERNAME=.*/DB_USERNAME=${DB_USERNAME:-glastree}/" .env
        sed -i "s/DB_PASSWORD=.*/DB_PASSWORD=${DB_PASSWORD:-secret}/" .env

        # Attende che MySQL sia pronto
        echo "Attendo MySQL..."
        until php -r "new PDO('mysql:host=${DB_HOST:-mysql};dbname=${DB_DATABASE:-glastree}', '${DB_USERNAME:-glastree}', '${DB_PASSWORD:-secret}');" 2>/dev/null; do
            sleep 2
        done
        echo "MySQL pronto."
    else
        # Configura SQLite
        sed -i 's/DB_CONNECTION=mysql/DB_CONNECTION=sqlite/' .env
        sed -i 's/DB_HOST=.*/' .env
        sed -i 's/DB_PORT=.*/' .env
        sed -i 's/DB_DATABASE=.*/DB_DATABASE=\var\www\html\database\database.sqlite/' .env
        sed -i 's/DB_USERNAME=.*/' .env
        sed -i 's/DB_PASSWORD=.*/' .env

        touch database/database.sqlite
        chmod 664 database/database.sqlite
    fi

    # Genera APP_KEY
    php artisan key:generate --force
fi

# Esegui migration e seed se il database è vuoto
if [ ! -f .db_initialized ]; then
    php artisan migrate --force
    php artisan db:seed --force
    touch .db_initialized
fi

# Configura storage link
php artisan storage:link --force 2>/dev/null || true

exec "$@"
```

```
chmod +x docker-entrypoint.sh
```

3. Seed Base Dati (comune)

Per includere tipologie documenti, tipologie eventi e ruoli, crea un seeder dedicato:

```
php artisan make:seeder DockerBaseDataSeeder
```

Contenuto di `database/seeder/DockerBaseDataSeeder.php` :

```
<?php

declare(strict_types=1);

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;

class DockerBaseDataSeeder extends Seeder
{
    public function run(): void
    {
        // Tipologie Documenti
        $tipologieDoc = ['documento', 'avatar', 'galleria', 'statuto', 'programma', 'locandina', 'altro'];
        foreach ($tipologieDoc as $ord => $nome) {
            DB::table('tipologie_documenti')->updateOrCreate(
                ['nome' => $nome],
                ['descrizione' => ucfirst($nome), 'ordine' => $ord + 1, 'attiva' => true]
            );
        }

        // Tipologie Eventi
        $tipologieEventi = [
            ['nome' => 'catechesi', 'descrizione' => 'Catechesi'],
            ['nome' => 'liturgia', 'descrizione' => 'Liturgia'],
            ['nome' => 'animazione', 'descrizione' => 'Animazione'],
            ['nome' => 'formazione', 'descrizione' => 'Formazione'],
            ['nome' => 'incontro', 'descrizione' => 'Incontro'],
            ['nome' => 'riunione', 'descrizione' => 'Riunione'],
            ['nome' => 'festa', 'descrizione' => 'Festa'],
            ['nome' => 'altro', 'descrizione' => 'Altro'],
        ];
        foreach ($tipologieEventi as $ord => $t) {
            DB::table('tipologie_eventi')->updateOrCreate(
                ['nome' => $t['nome']],
                ['descrizione' => $t['descrizione'], 'ordine' => $ord + 1, 'attiva' => true]
            );
        }

        // Ruoli
        $ruoli = ['presidente', 'vicepresidente', 'segretario', 'tesoriere', 'consigliere', 'volontario',
'socio', 'altro'];
        foreach ($ruoli as $ord => $nome) {
            DB::table('ruoli')->updateOrCreate(
                ['nome' => $nome],
                ['descrizione' => ucfirst($nome), 'ordine' => $ord + 1, 'attiva' => true]
            );
        }
    }
}
```

Registralo in `database/seeder/DatabaseSeeder.php` :

```
$this->call([
    // ... altri seeder ...
    DockerBaseDataSeeder::class,
]);
```

4. Docker Compose

Opzione A — SQLite (leggero)

```
services:
  glastree:
    build: .
    ports:
      - "8080:80"
    volumes:
      - glastree_storage:/var/www/html/storage
      - glastree_db:/var/www/html/database
    environment:
      - APP_ENV=production
      - APP_DEBUG=false
      - APP_URL=http://localhost:8080
      - DB_CONNECTION=sqlite

volumes:
  glastree_storage:
  glastree_db:
```

Opzione B — MySQL (standard)

```

services:
  glastree:
    build: .
    ports:
      - "8080:80"
    volumes:
      - glastree_storage:/var/www/html/storage
    depends_on:
      mysql:
        condition: service_healthy
    environment:
      - APP_ENV=production
      - APP_DEBUG=false
      - APP_URL=http://localhost:8080
      - DB_CONNECTION=mysql
      - DB_HOST=mysql
      - DB_PORT=3306
      - DB_DATABASE=glastree
      - DB_USERNAME=glastree
      - DB_PASSWORD=secret

  mysql:
    image: mysql:8.0
    volumes:
      - mysql_data:/var/lib/mysql
    environment:
      - MYSQL_DATABASE=glastree
      - MYSQL_USER=glastree
      - MYSQL_PASSWORD=secret
      - MYSQL_ROOT_PASSWORD=root_secret
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      interval: 5s
      timeout: 3s
      retries: 10

volumes:
  glastree_storage:
  mysql_data:

```

5. Build e avvio

Assicurati di essere nella cartella del progetto (ottenuta via `git clone` o estraendo il ZIP), poi:

```

# Costruisci l'immagine (comune)
docker build -t glastree:latest .

# Avvia con SQLite
docker compose -f docker-compose.yml up -d

# Oppure avvia con MySQL
# docker compose -f docker-compose.mysql.yml up -d

# Apri nel browser
open http://localhost:8080

```

6. Primo accesso

Al primo avvio, l'entrypoint esegue automaticamente migration e seed. Troverai già popolati:

- **Tipologie Documenti:** documento, avatar, galleria, statuto, programma, locandina, altro
- **Tipologie Eventi:** catechesi, liturgia, animazione, formazione, incontro, riunione, festa, altro
- **Ruoli:** presidente, vicepresidente, segretario, tesoriere, consigliere, volontario, socio, altro

Dovrai creare il primo utente amministratore via CLI:

```
docker exec -it glastree php artisan tinker --execute="\$u = new \\App\\Models\\User;
\\$u->name = 'Amministratore';
\\$u->email = 'admin@esempio.it';
\\$u->password = bcrypt('password-forte');
\\$u->is_admin = true;
\\$u->permissions = ['indivui' => 2, 'gruppi' => 2, 'eventi' => 2, 'documenti' => 2, 'mailing' => 2, 'viste' => 2, 'report' => 2, 'settings' => 2];
\\$u->save();
echo 'Utente creato con successo!';
"
```

Note importanti

- **Persistenza:** i volumi Docker preservano storage e database tra i riavvii
- **APP_KEY:** viene generata al primo avvio — se elimini i volumi, la perdi e i dati criptati diventano inaccessibili
- **MySQL in produzione:** imposta password sicure per `MYSQL_PASSWORD` e `MYSQL_ROOT_PASSWORD`
- **File compose separati:** puoi mantenere entrambi i file (`docker-compose.yml` per SQLite, `docker-compose.mysql.yml` per MySQL) e scegliere con `-f`