

# Guide de Configuration Manuelle de la Base de Données PostgreSQL sur Railway

---

## Prérequis

- Un compte Railway
- L'application EVIMERIA déjà déployée sur Railway
- Accès à la ligne de commande Railway (CLI)

## Étapes de Configuration Manuelle

### 1. Installation de la CLI Railway

```
# Installation via npm
npm i -g @railway/cli

# Connexion à votre compte
railway login
```

### 2. Création de la Base de Données

1. Dans l'interface Railway, allez dans votre projet
2. Cliquez sur "New"
3. Sélectionnez "Database" puis "PostgreSQL"
4. Attendez la création de la base de données

### 3. Configuration des Variables d'Environnement

Les variables suivantes seront automatiquement créées par Railway :

```
DATABASE_URL=postgresql://${PGUSER}:${POSTGRES_PASSWORD}@${RAILWAY_TCP_PROXY_DOMAIN}:${RAILWAY_TCP_PROXY_PORT}/${PGDATABASE}
PGDATABASE=railway
PGHOST=containers-us-west-XX.railway.app
PGPASSWORD=votre_mot_de_passe_généré
PGPORT=5432
PGUSER=postgres
POSTGRES_DB=railway
POSTGRES_PASSWORD=votre_mot_de_passe_généré
POSTGRES_USER=postgres
```

### 4. Connexion à la Base de Données

```
# Connexion via psql
railway connect

# OU en utilisant l'URL complète
psql "postgresql://$PGUSER:$PGPASSWORD@$PGHOST:$PGPORT/$PGDATABASE"
```

## 5. Exécution des Migrations

```
# Se placer dans le répertoire backend
cd backend

# Exécuter les migrations
railway run python manage.py migrate

# Créer un superutilisateur
railway run python manage.py createsuperuser
```

## 6. Vérification de la Base de Données

```
-- Connexion à la base
\c railway

-- Liste des tables
\dt

-- Vérification des utilisateurs
SELECT * FROM users_user;

-- Vérification des catégories
SELECT * FROM products_category;
```

## Commandes Utiles

### Sauvegarde de la Base de Données

```
# Création d'une sauvegarde
railway run pg_dump > backup.sql

# Restauration d'une sauvegarde
railway run psql < backup.sql
```

### Réinitialisation de la Base de Données

```
# Suppression de toutes les tables
railway run python manage.py flush

# Recréation des tables
railway run python manage.py migrate
```

## Résolution des Problèmes Courants

### 1. Erreur de Connexion

Si vous rencontrez une erreur "permission denied" :

- Vérifiez que les variables d'environnement sont correctement configurées
- Assurez-vous que l'adresse IP est autorisée dans les règles de pare-feu Railway

### 2. Erreur de Migration

Si les migrations échouent :

```
# Réinitialiser les migrations
railway run python manage.py migrate --fake
railway run python manage.py migrate --fake-initial
railway run python manage.py migrate
```

### 3. Erreur SSL

Si vous avez des erreurs SSL :

- Assurez-vous que `sslmode=require` est dans l'URL de connexion
- Vérifiez que le certificat SSL est valide

## Monitoring

### Surveillance des Performances

Dans l'interface Railway :

1. Allez dans l'onglet "Metrics"
2. Surveillez :
  - L'utilisation CPU
  - La mémoire utilisée
  - Les connexions actives
  - L'espace disque utilisé

### Logs de la Base de Données

```
# Afficher les logs en temps réel
railway logs

# Filtrer les logs PostgreSQL
railway logs | grep postgres
```

## Sécurité

### Bonnes Pratiques

1. Ne jamais exposer les identifiants de base de données
2. Utiliser des mots de passe forts
3. Limiter les accès aux adresses IP nécessaires
4. Activer le chiffrement SSL
5. Sauvegarder régulièrement la base de données

### Configuration du Pare-feu

Dans Railway :

1. Allez dans les paramètres de la base de données
2. Section "Networking"
3. Configurez les règles IP autorisées

## Optimisation

### Index et Performance

```
-- Création d'index pour les requêtes fréquentes
CREATE INDEX idx_product_name ON products_product(name);
CREATE INDEX idx_category_slug ON products_category(slug);

-- Analyse des performances
EXPLAIN ANALYZE SELECT * FROM products_product WHERE category_id = 1;
```

### Maintenance

```
-- Nettoyage et optimisation
VACUUM ANALYZE;

-- Reconstruction des index
REINDEX TABLE products_product;
```