

Documentation de Déploiement - EcoRide

Ce guide détaille la procédure suivie pour déployer l'application EcoRide sur l'environnement de production **Railway**.

1. Démarche et Infrastructure

Nous avons choisi une architecture moderne et automatisée pour le code, tout en gardant un contrôle manuel sur la base de données pour plus de sécurité et de précision.

- **Hébergement (PaaS) : Railway** (Pour sa simplicité de lien avec GitHub et PHP natif).
 - **Base de Données : MySQL** (Hébergée sur Railway, gérée via HeidiSQL).
 - **Déploiement Code : Automatique** (Push sur la branche `main` de GitHub -> Build Railway).
 - **Initialisation Données : Manuelle** (Import SQL via client lourd + saisie back-office).
-

2. Prérequis Techniques

Pour reproduire ce déploiement, il faut :

1. Un dépôt **GitHub** contenant le projet à jour.
 2. Un compte **Railway**.
 3. Le logiciel **HeidiSQL** (ou DBeaver) installé en local pour gérer la BDD à distance.
 4. Les fichiers de configuration racine (`composer.json`, `Procfile`) présents dans le dépôt.
-

3. Étapes de Mise en Production

Étape A : Création du Service Web (Railway)

1. Sur Railway, création d'un **Nouveau Projet**.
2. Sélection de "**Deploy from GitHub repo**".
3. Choix du dépôt **EcoRide**.
4. Railway détecte automatiquement le `composer.json` et lance le conteneur PHP.
5. Le site est en ligne mais affiche une erreur SQL (car la BDD n'est pas encore liée).

Étape B : Base de Données & Variables

1. Dans le projet Railway, ajout d'un service **MySQL**.
2. Récupération des identifiants (Host, User, Password, Port) dans l'onglet "Connect".
3. Dans le service Web (PHP) -> Onglet **Variables**, ajout des clés pour la connexion :
 - `MYSQLHOST`
 - `MYSQLPORT`
 - `MYSQLDATABASE`
 - `MYSQLUSER`
 - `MYSQLPASSWORD`
4. Redémarrage du service Web pour prise en compte.

Étape C : Initialisation de la BDD (Via HeidiSQL)

Contrairement à une approche tout automatisée, nous avons choisi d'utiliser un client SQL pour vérifier chaque étape de la création.

1. Ouverture de **HeidiSQL** sur le poste local.
2. Création d'une nouvelle session avec les identifiants Railway (Host TCP/IP, User, Password).
3. Connexion réussie à la base distante.
4. **Import de la structure :**
 - Ouverture du fichier `sql/schema.sql` dans HeidiSQL.
 - Exécution du script (F9).
 - *Résultat* : Les tables (`utilisateur`, `covoiturage`, `avis...`) sont créées et les comptes administrateurs par défaut (Admin, Employé) sont insérés (via les `INSERT` présents dans le schéma).

Étape D : Remplissage des Données (Données de test)

Pour s'assurer de la cohérence des trajets, les données ont été saisies manuellement :

1. Connexion au site déployé avec le compte **Admin** ou **Conducteur** (créé par le script).
 2. Publication manuelle de trajets réels (Paris-Lyon, Lyon-Marseille...) via le formulaire "Publier un trajet".
 3. Vérification dans HeidiSQL que les nouvelles entrées apparaissent bien en base.
-

4. Vérification Finale

1. Accès à l'URL publique (ex: <https://ecoride-production.up.railway.app>).
2. Test de connexion (Login OK).
3. Recherche d'un trajet (Les trajets saisis manuellement apparaissent).
4. Le déploiement est **validé**.