

# Documentation de déploiement - Application EcoRide

## Table des matières

1. Introduction
2. Prérequis
3. Configuration de la base de données sur Heroku
4. Migration des données
5. Configuration de l'application
6. Déploiement de l'application
7. Vérification et tests
8. Remarques importantes

## 1. Introduction

Ce document décrit la procédure de déploiement de l'application EcoRide sur la plateforme Heroku. EcoRide est une application de covoiturage qui nécessite une base de données MySQL pour fonctionner correctement.

## 2. Prérequis

- Un compte Heroku
- Git installé localement
- MySQL installé localement
- L'application EcoRide fonctionnelle en environnement local

## 3. Configuration de la base de données sur Heroku

Heroku ne prend pas en charge MySQL nativement. Pour résoudre ce problème, j'ai utilisé l'extension JawsDB qui permet d'héberger une base de données MySQL sur Heroku.

bash

```
# Création de l'extension JawsDB avec le plan kitefin
heroku addons:create jawsdb:kitefin
```

```
# Vérification du statut de la base de données
heroku addons:info jawsdb-spherical-15054
```

Résultat obtenu :

```
=== jawsdb-spherical-15054
Attachments:  studiecoride::JAWSDB
Installed at: Mon Mar 24 2025 12:38:39 GMT+0100 (Central European
Standard Time)
Max Price:    free
Owning app:   studiecoride
Plan:        jawsdb:kitefin
Price:       free
State:       created
```

J'ai récupéré les informations de connexion :

bash

```
heroku config | grep JAWSDB_URL
```

Les informations de connexion sont au format suivant  
mysql://username:password@hostname:port/database\_name :

```
mysql://zunyfzr1adh6n4l6:b8i4ygwks4dnme1q@nba02whlntki5w2p.cbetxkd
yhwsb.us-east-1.rds.amazonaws.com:3306/zyi88je2y9imrqzc
```

Dans mon cas :

- Hostname : nba02whlntki5w2p.cbetxkdyhwsb.us-east-1.rds.amazonaws.com
- Username : zunyfzr1adh6n4l6
- Password : b8i4ygwks4dnme1q
- Database Name : zyi88je2y9imrqzc
- Port : 3306

## 4. Configuration de l'application

J'ai modifié le fichier de configuration de la base de données pour supporter à la fois l'environnement local et Heroku :

php

```
<?php
// Obtention de l'URL de la base de données depuis la variable
d'environnement
$dbUrl = getenv('JAWSDB_URL');
if ($dbUrl) {
    // Analyse de l'URL en composants
    $dbParts = parse_url($dbUrl);
    $host = $dbParts['host'];
    $username = $dbParts['user'];
    $password = $dbParts['pass'];
    $dbname = ltrim($dbParts['path'], '/');
```

```

        $port = isset($dbParts['port']) ? $dbParts['port'] : 3306;
    } else {
        // Paramètres locaux pour le développement
        $host = 'localhost';
        $dbname = 'DB_EcoRide';
        $username = 'root';
        $password = 'Molfarka8';
        $port = '3306';
    }
    try {
        // Création de la connexion
        $pdo = new PDO(
            "mysql:host=$host;port=$port;dbname=$dbname;charset=utf8mb4",
            $username,
            $password,
            [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
                PDO::ATTR_EMULATE_PREPARES => false
            ]
        );
    } catch (PDOException $e) {
        die("Erreur de connexion à la base de données : " .
        $e->getMessage());
    }
}

```

## 5. Migration des données

Les fournisseurs d'hébergement comme Heroku créent des utilisateurs de base de données avec des privilèges limités (pas de privilège SUPER). Pour éviter les erreurs liées aux privilèges, j'ai exporté séparément la structure et les données :

bash

```

# Export de la structure sans les données
mysqldump -u root -pMolfarka8 --no-data --skip-triggers DB_EcoRide
> ecoride_structure.sql

```

```

# Export des données sans la structure
mysqldump -u root -pMolfarka8 --no-create-info --skip-triggers
--complete-insert DB_EcoRide > ecoride_data.sql

```

**Important:** J'ai édité le fichier ecoride\_structure.sql pour supprimer toutes les lignes contenant DEFINER et GTID.

J'ai importé la structure et les données dans JawsDB :

bash

```
# Import de la structure
mysql -h nba02whlntki5w2p.cbetxkdyhwsb.us-east-1.rds.amazonaws.com
-u zunyfhzrladh6n4l6 -pb8i4ygwks4dnmelq zyi88je2y9imrqzc <
ecoride_structure.sql
```

```
# Import des données
mysql -h nba02whlntki5w2p.cbetxkdyhwsb.us-east-1.rds.amazonaws.com
-u zunyfhzrladh6n4l6 -pb8i4ygwks4dnmelq zyi88je2y9imrqzc <
ecoride_data.sql
```

J'ai vérifié l'importation :

bash

```
mysql -h nba02whlntki5w2p.cbetxkdyhwsb.us-east-1.rds.amazonaws.com
-u zunyfhzrladh6n4l6 -pb8i4ygwks4dnmelq zyi88je2y9imrqzc -e "SHOW
TABLES;"
```

Résultat obtenu :

```
+-----+
| Tables_in_zyi88je2y9imrqzc |
+-----+
| avis                        |
| conducteur_ratings         |
| configuration              |
| covoiturage                |
| disponible_covoiturations  |
| eco_covoiturations         |
| marque                     |
| parametre                  |
| participation               |
| role                        |
| utilisateur                |
| voiture                    |
+-----+
```

J'ai également vérifié le contenu de la table des rôles :

```
+-----+-----+
| role_id | libelle      |
+-----+-----+
| 1       | utilisateur  |
| 2       | passager     |
| 3       | chauffeur    |
| 4       | employe     |
| 5       | administrateur |
```

+-----+-----+

Ces résultats m'ont confirmé que toutes les tables et données avaient été correctement importées dans la base de données JawsDB sur Heroku.

## 6. Déploiement de l'application

Une fois la base de données configurée et le fichier de configuration mis à jour, j'ai déployé l'application :

bash

```
git add config/database.php
git commit -m "Configuration DB pour Heroku"
git push heroku main
```

## 7. Vérification et tests

Après le déploiement, j'ai vérifié que l'application fonctionnait correctement sur Heroku :

- J'ai vérifié que toutes les pages s'affichaient correctement
- J'ai testé les fonctionnalités principales (inscription, connexion, ajout de voiture, etc.)
- J'ai vérifié les journaux d'erreurs pour m'assurer qu'il n'y avait pas de problèmes

## 8. Remarques importantes

**Attention** : Les bases de données locales et sur Heroku ne se synchronisent pas automatiquement. J'ai constaté que la base de données locale contient les données créées pendant mes tests sur le serveur local, tandis que la base de données JawsDB sur Heroku accumule les données de "production", c'est-à-dire de la version du site accessible aux utilisateurs réels.

Pour mettre à jour la base de données de production après des modifications locales, je dois répéter le processus d'exportation et d'importation.