

Grandes Lignes du Rapport de la Procédure de Déploiement

Voici une version condensée et explicative de la procédure que nous allons inclure dans notre rapport de déploiement:

Rapport de Déploiement : BusinessBook (Next.js + PostgreSQL)

Objectif : Déployer l'application web "BusinessBook", développée avec Next.js, TailwindCSS et Prisma, en utilisant Vercel pour l'hébergement de l'application et Neon pour la base de données PostgreSQL.

Outils utilisés :

- **Vercel** : Plateforme de déploiement front-end (pour Next.js).
- **Neon** : Base de données PostgreSQL sans serveur.
- **GitHub** : Dépôt de code source.
- **pnpm** : Gestionnaire de paquets (utilisé pour les dépendances Next.js).
- **Prisma** : ORM pour la gestion de la base de données.

Étapes du Déploiement :

1. Configuration de la Base de Données PostgreSQL sur Neon :
 - Création d'un compte sur [Neon.tech](#).
 - Initialisation d'un nouveau projet de base de données PostgreSQL via le tableau de bord Neon.
 - Récupération de la **chaîne de connexion (Connection String)** complète fournie par Neon, essentielle pour la connexion de l'application.
 - (*Optionnel si des données initiales sont requises*) Migration des données existantes de la base de données PostgreSQL locale (pgAdmin) vers la base de données Neon via des outils comme DBeaver (export/import SQL).
2. Préparation du Projet Next.js (Localement) :
 - **Mise à jour des Variables d'Environnement** : Création ou modification du fichier .env.local à la racine du projet. Ce fichier contient des informations sensibles non versionnées :
 - **DATABASE_URL** : Assignation de la chaîne de connexion Neon récupérée.
 - **AUTH_SECRET** : Définition d'une clé secrète pour l'authentification (Next-Auth).
 - **Configuration de Prisma** : Vérification du fichier prisma/schema.prisma pour s'assurer que la source de données (datasource db) utilise la variable d'environnement DATABASE_URL.

- **Génération du Client Prisma et Poussée du Schéma** : Exécution des commandes `pnpm prisma generate` et `pnpm prisma db push` dans le terminal pour générer le client ORM et appliquer le schéma de base de données à la base de données Neon.
- **Vérification Git** : Confirmation que le fichier `.env.local` est bien listé dans `.gitignore` pour ne pas être poussé sur le dépôt public.
- **Commit et Push Git** : Soumission des dernières modifications du code source (sauf `.env.local`) au dépôt GitHub du projet.

3. Déploiement sur Vercel :

- **Connexion à Vercel** : Accès au tableau de bord Vercel via authentification GitHub.
- **Importation du Dépôt** : Sélection du dépôt `businessbook` depuis la liste des dépôts GitHub disponibles dans l'interface Vercel.
- **Configuration du Projet Vercel** :
 - Détection automatique du **Framework Next.js**.
 - Confirmation des **commandes de build et d'installation** (Vercel gère automatiquement `pnpm build` et `pnpm install`).
 - **Ajout Crucial des Variables d'Environnement** : Saisie manuelle de `DATABASE_URL` et `AUTH_SECRET` (avec leurs valeurs respectives) dans la section "**Environment Variables**" des paramètres du projet Vercel. Ces variables sont injectées au moment du build et de l'exécution sur les serveurs de Vercel.
- **Initiation du Déploiement** : Lancement du processus de déploiement via le bouton "Deploy". Vercel récupère le code, installe les dépendances, effectue la construction (build) et déploie l'application.
- **Validation** : Accès à l'application via l'URL générée par Vercel après un déploiement réussi, confirmant sa disponibilité et sa fonctionnalité en ligne.