

# Mise en prod du Backend de la Todolist

En essayant de mettre en prod le backend de notre application, nous nous sommes rapidement aperçu que la technologie de base de données que nous utilisions [n'était pas compatible avec Vercel](#). Cependant, le même article nous propose rapidement une solution avec un SGBD intégré à Vercel : [Postgresql](#), mais c'est alors qu'un autre challenge se profile à l'horizon : Sequelize n'est pas encore supporté par Vercel, ce qui nous indique que l'installation et l'utilisation de cet ORM sera un peu plus compliquée que prévu. Let's go !

## Configuration nécessaire pour Neon

Avant de commencer à modifier notre code, nous allons dans un premier temps avoir une nouvelle BDD fonctionnant sous Postgresql. Fort heureusement, nous avons déjà par le passé travaillé avec Neon, un SaaS qui héberge des BDD Postgres et nous allons le réutiliser.

Nous pouvons pour cela créer une nouvelle base de données, **se sera votre 1ère tâche**

Une fois faites, vous devriez arriver sur le dashboard suivant :

The screenshot shows the Neon dashboard interface. At the top, there's a breadcrumb 'Home / backend\_test' and a 'FREE Upgrade to Pro' button. The main content is divided into four panels:

- Connection Details:** Shows 'Branch' as 'main' (PRIMARY), 'Compute' as 'RW' (IDLE), 'Database' as 'neondb', and 'Role' as 'Niitok'. It includes a 'Reset password' link and a 'Pooled connection' checkbox. A connection string is displayed: 'postgres://Niitok:\*\*\*\*\*@ep-sweet-lake-23223658.eu-central-1.aws.neon.tech/neondb'. A note at the bottom says 'Your password is saved in a secure storage vault.'
- Branches:** A table with columns 'Name', 'RW compute', and 'Used space'. It shows one branch 'main' (PRIMARY) with '0.25 CU' and '0%' used space.
- Usage since September 12:** Shows 'Active time all computes' as '0%' and '0.36 h / 100 h'. A note states: 'The Free Tier grants 100 hours of Active time per month for all computes. Exceeding this limit, non-primary branch computes are subject to suspension. The primary branch compute remains available.' A link 'To increase project limits, upgrade to Pro' is present.
- Storage:** Shows 'Project storage' as '31 MiB' and 'Branches' as '1 / 10'.

Sur ce dashboard, vous trouverez le nom d'utilisateur, le mot de passe, le nom de la BDD et son adresse.

Attention cependant, Neon n'autorise que [les connections sécurisées aux BDD](#) que le service héberge. En vous essayant de vous y connecter, vous tomberez rapidement sur une erreur ssl. Il faudra donc trouver une solution pour gérer ce problème lors de la configuration de la connexion avec Sequelize (cf Sources).

## Configuration nécessaire pour Sequelize

Ensuite, il faudra modifier la configuration de notre dépôt et de notre ORM pour pouvoir entamer notre migration. Dans notre fichier de configuration de Sequelize, nous devons d'abord changer le gestionnaire de base de données de notre service. Ce code devrait être nécessaire pour pouvoir configurer à minima notre base de données. Toutes les informations nécessaires se trouvant sur le dashboard de notre base de données hébergé par Neon.

```
const sequelize = new Sequelize(DB.NAME, DB.USER, DB.USER.PWD, {
  host: DB.ADDR,
```

```
dialect: "postgres", // cf documentation de Sequelize
});
```

C'est aussi à cet endroit qu'il faudra modifier les paramètres *dialectOptions* pour autoriser la connection SSL à votre BDD.

et il ne faudra pas oublier d'installer les paquets "pg" et "pg-hstore" avec npm pour pouvoir utiliser la BDD, en plus de supprimer le paquet désormais inutile sqlite.

```
npm install pg pg-hstore
```

## Configuration nécessaire pour Vercel

Vous devriez être assez avancés pour lancer votre serveur et essayer vos premières requêtes sur la BDD. Essayez avec le fichier .http de faire quelques get et post sur le serveur.

Après un commit et un push sur votre dépôt github, vous devriez pouvoir mettre votre projet sur Vercel. Et là, rien ne marche...

Il existe plusieurs raisons à cela, mais c'est principalement dû au manque d'intégration de Node et de ts à Vercel. Il faudra en effet un peu adapter son dépôt en suivant les consignes de [ce post](#).

La seconde erreur se déclare pendant la compilation

l'erreur "Please install pg package manually"

Cette erreur est assez simple à résoudre. Il s'agit d'un import défectueux dans Vercel du paquet de postgres. [La discussion suivante](#) aide à le résoudre avec un ajout d'une ligne de code. Les explications sur cette erreur se trouve quant à elles sur [cette page](#), cette erreur s'explique principalement par la configuration et l'utilisation de [fonctions serverless](#) de Vercel.

## La suite : Les variables d'environnement

### Sources

<https://vercel.com/guides/is-sqlite-supported-in-vercel>

<https://vercel.com/docs/storage/vercel-postgres>

<https://neon.tech/docs/connect/connect-securely>

<https://dev.to/tirthpatel/deploy-node-ts-express-typescript-on-vercel-284h>

<https://community.neon.tech/t/how-to-configure-sslmode-in-sequelize/767>

<https://sequelize.org/docs/v6/getting-started/#connecting-to-a-database>

<https://vercel.com/guides/using-express-with-vercel>

<https://github.com/orgs/vercel/discussions/234>

[https://github.com/Niirok/back\\_todo](https://github.com/Niirok/back_todo) à forker pour prendre en exemple back à sable

<https://github.com/Niirok/express-typescript-vercel-part2> idem