



**IT CONSULTING**  
& Development

# OC Pizza

## **Système de gestion de pizzéria**

Dossier d'exploitation

Version 1.0

### **Auteur**

Vincent Caronnet  
*Software Engineer*

**{}{{Entreprise}}D**  
Itcd.fr

168, bd. de Magenta – 75010 Paris – 01 22 33 44 55 66 – itcd.fr  
S.A.R.L. au capital de 100 000,00 € enregistrée au RCS de Paris  
SIREN 999 999 999 – Code APE : 6202A

## TABLE DES MATIERES

<b>1 -Versions .....</b>	<b>4</b>
<b>2 -Introduction .....</b>	<b>5</b>
2.1 -Objet du document.....	5
2.2 -Références.....	5
<b>3 -Pré-requis .....</b>	<b>6</b>
3.1 -Système .....	6
3.1.1 -Serveur de Base de données.....	6
3.1.1.1 -Caractéristiques techniques .....	6
3.1.2 -Serveur Web.....	6
3.1.2.1 -Caractéristiques techniques .....	6
3.1.3 -Serveur de Batches.....	6
3.1.3.1 -Caractéristiques techniques .....	6
3.1.4 -Serveur de Fichiers.....	6
3.2 -Bases de données .....	6
3.3 -Web-services .....	6
3.4 -Autres Ressources.....	6
<b>4 -Procédure de déploiement.....</b>	<b>7</b>
4.1 -Déploiement des Batches .....	7
4.1.1 -Artefacts .....	7
4.1.2 -Variables d'environnement.....	7
4.1.3 -Configuration.....	7
4.1.3.1 -Fichier xxx.yyy .....	8
4.1.3.2 -Fichier zzz.ttt.....	8
4.1.3.3 -Fichier .....	8
4.1.4 -Ressources .....	8
4.1.5 -Vérifications .....	8
4.2 -Déploiement de l'Application Web.....	9
4.2.1 -Artefacts .....	9
4.2.2 -Environnement de l'application web .....	9
4.2.2.1 -Variables d'environnement.....	9
4.2.3 -Répertoire de configuration applicatif .....	9
4.2.3.1 -Fichier xxx.yyy .....	9
4.2.4 -DataSources.....	9
4.2.5 -Ressources .....	10
4.2.6 -Vérifications .....	10
<b>5 -Procédure de démarrage / arrêt.....</b>	<b>11</b>
5.1 -Base de données .....	11
5.2 -Batches .....	11
5.3 -Application web .....	11
<b>6 -Procédure de mise à jour.....</b>	<b>12</b>
6.1 -Base de données .....	12

6.2 -Batches .....	12
6.3 -Application web .....	12
<b>7 -Supervision/Monitoring.....</b>	<b>13</b>
7.1 -Supervision de l'application web.....	13
<b>8 -Procédure de sauvegarde et restauration.....</b>	<b>14</b>
<b>9 -Glossaire .....</b>	<b>15</b>

## 1 - VERSIONS

Auteur	Date	Description	Version
Vincent Caronnet	29/05/2021	Création du document	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception technique du système de gestion de pizzéries développé par IT Consulting & Development pour OC Pizza découlant :

1. des besoins d'OC Pizza recueillis le 3/05/2021 par IT Consulting & Development.
2. du Dossier de conception fonctionnelle produit suite à ce recueil des besoins,
3. du Dossier de conception technique.

Dans le cadre des méthodes Agile utilisées pour le développement, la mise en production et la maintenance du système, ce document devra être mis à jour en fonction de l'évolution du projet, de ses nouveaux enjeux et des nouvelles solutions qui seront mises en œuvre dans le futur.

Chaque mise à jour de ce document par le prestataire devra être à nouveau vérifiée et approuvée par le client.

Les versions successives de ce document seront conservées et versionnées dans ce dépôt GitHub : <https://github.com/centvingt/OCPizzaRedaction>.

### 2.2 - Références

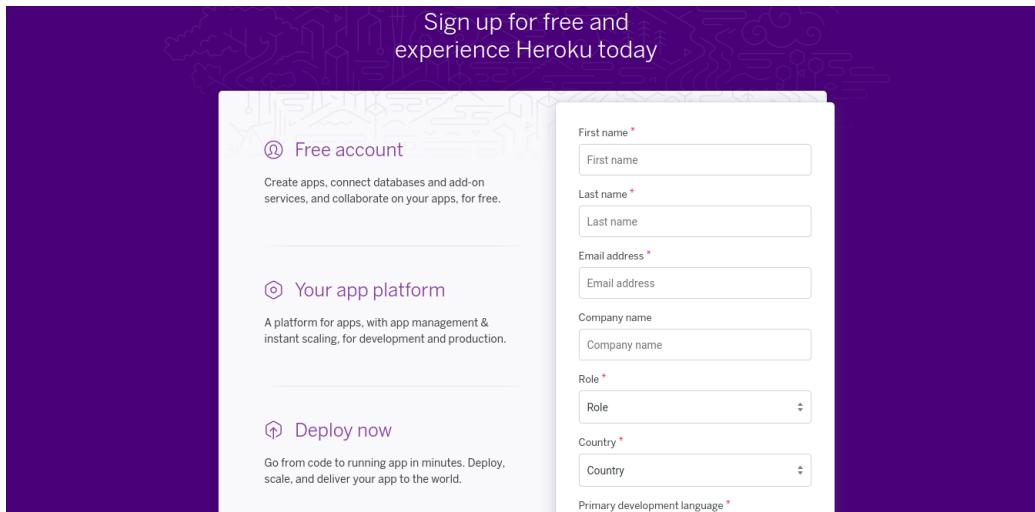
Pour de plus amples informations, se référer également aux éléments suivants :

1. Dossier de conception fonctionnelle du système
2. Dossier de conception technique du système

## 3 - PRÉ-REQUIS

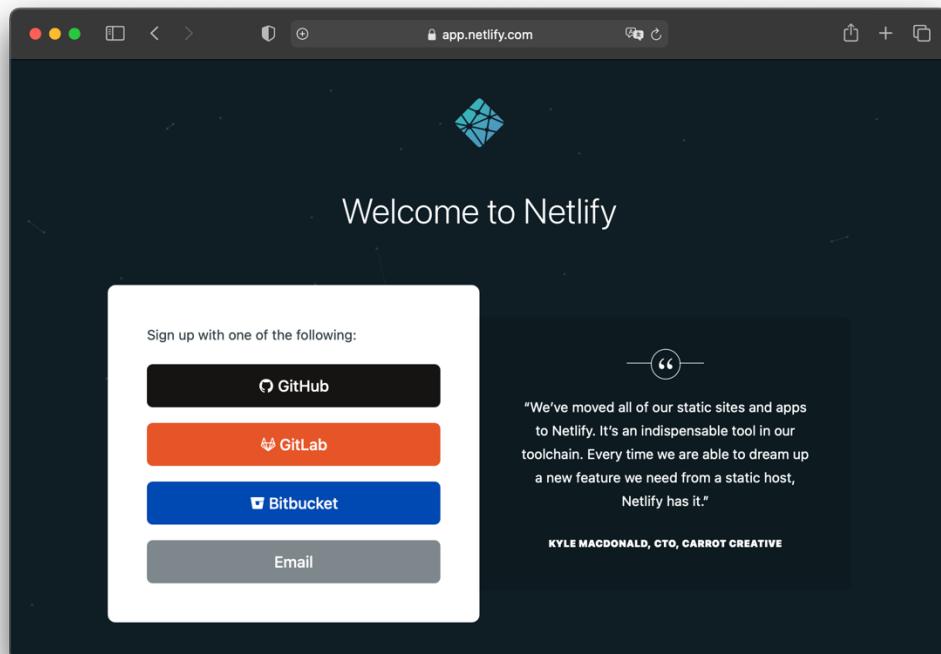
### 3.1 - Compte Heroku

L'API Node.js et la base de données PostgreSQL sont hébergées par Heroku chez qui il est nécessaire de créer un compte.



### 3.2 - Compte Netlify

De même, il est nécessaire de créer un compte chez Netlify pour y déployer le site statique.



### 3.3 - Applications mobiles

Les applications mobiles, développées avec React Native, nécessitent l'installation de cette librairie ainsi que de la plateforme de développement Expo. La procédure d'installation de ces outils est détaillée sur cette page : <https://reactnative.dev/docs/environment-setup>

De plus, il est nécessaire de créer des comptes de développeur sur l'App Store (iOS) et sur Play Store (Android).

Par ailleurs, un Mac tournant avec Big Sur (macOS 12) avec le logiciel gratuit Xcode est nécessaire pour la publication des applications mobiles sur l'App Store.

La publication sur le Play Store nécessite l'utilisation du logiciel gratuit Android Studio.

### 3.4 - Dépôts GitHub

Le groupe OC Pizza a créé le compte **@ocpizza** sur GitHub, les fichiers sources des projets du système développé sont partagés avec ce compte sur les dépôts suivants :

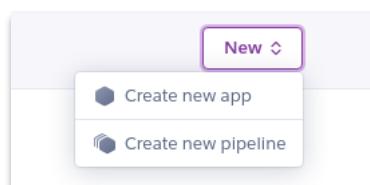
- <https://github.com/ocpizza/api.git> pour l'API,
- <https://github.com/ocpizza/webapp.git> pour le site statique,
- <https://github.com/ocpizza/customer-app.git> pour l'application mobile du client de la pizzéria,
- <https://github.com/ocpizza/pizzamaker-app.git> pour l'application mobile du pizzaïolo,
- <https://github.com/ocpizza/deleverman-app.git> pour l'application mobile du livreur.

## 4 - PROCÉDURES DE DÉPLOIEMENTS ET DE PUBLICATIONS

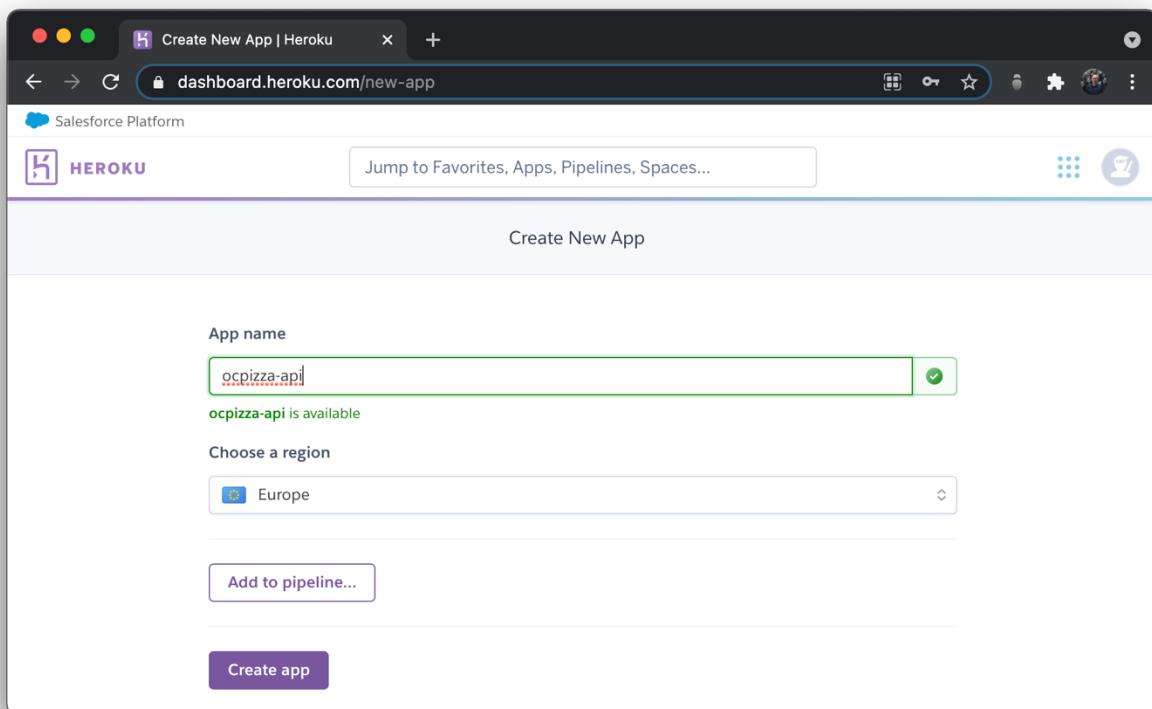
### 4.1 - Déploiement de l'API et de sa base de données sur Heroku

#### 4.1.1 - Ajout de l'application au compte Heroku

Se connecter au tableau de bord d'Heroku et créer une nouvelle application :



Donner un nom à cette application, sélectionner la région et cliquer sur **Create app** :





Une fois que l'application est créée, cliquer sur le bouton **GitHub** en bas de la page, puis cliquer sur le bouton **Connect to GitHub** :

The screenshot shows the Heroku Dashboard for the app 'ocpizza-api'. At the top, there's a banner for 'Salesforce Platform'. Below it, the Heroku navigation bar includes 'Personal', 'ocpizza-api', 'Open app', and 'More'. The main content area has tabs for 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. Under 'Deploy', there are sections for adding the app to a pipeline and connecting it to GitHub. The 'Deployment method' section shows options for Heroku Git (using Heroku CLI), GitHub (using GitHub CLI), and Container Registry (using Heroku CLI). The 'Connect to GitHub' section allows users to connect their app to a GitHub repository. A large purple button labeled 'Connect to GitHub' is prominently displayed at the bottom of this section.

Une fois que votre compte Heroku est connecté à votre compte GitHub, recherchez le dépôt GitHub de l'API :

This screenshot shows the 'Connect to GitHub' interface on the Heroku Dashboard. It features a search bar where 'centvingt' is selected from a dropdown and 'ocpizza-api' is typed into the search field. A purple 'Search' button is to the right. Below the search bar, a message says 'Missing a GitHub organization? [Ensure Heroku Dashboard has team access](#)'. At the bottom, a list shows 'centvingt/ocpizza-api' with a 'Connect' button to its right.



Après avoir cliqué sur le bouton **Connect**, l'API est déployée :

The screenshot shows the Heroku dashboard for the OC Pizza API. Under 'Deployment method', there are three options: 'Heroku Git' (using Heroku CLI), 'GitHub Connected' (with a green checkmark), and 'Container Registry' (using Heroku CLI). Below this, a section titled 'App connected to GitHub' indicates the app is connected to centvingt/ocpizza-api by centvingt. It also mentions that code diffs, manual and auto deploys are available. A note says 'Releases in the [activity feed](#) link to GitHub to view commit diffs'. There is a 'Disconnect...' button.

L'API est maintenant accessible à l'URL <https://ocpizza-api.herokuapp.com>, il est alors possible de lui fournir un nom de domaine dans l'onglet **Settings** :

The screenshot shows the 'Domains' section in the Heroku app settings. It says 'Your app can be found at <https://ocpizza-api.herokuapp.com/>' and has a 'Add domain' button. A search bar is labeled 'Filter domains'. A note says 'Custom domains will appear here' and 'Custom domains allow you to access your app via one or more non-Heroku domain names (for example, www.yourcustomdomain.com)'.

#### 4.1.1.1 - Ajout de la base de données PostgreSQL à l'application dans Heroku

Aller dans l'onglet **Resources** et rechercher **Heroku Postgres**

The screenshot shows the 'Resources' page in the Heroku dashboard. It has tabs for Overview, Resources (which is selected), Deploy, Metrics, Activity, Access, and Settings. Under 'Dynos', it says 'This app has no process types yet' and 'Add a Procfile to your app in order to define its process types.' Under 'Add-ons', a search bar shows 'heroku postgres'. A result for 'Heroku Postgres' is listed with the note 'There are no add-ons for this app' and 'You can add add-ons to this app and they will show here.' At the bottom, it shows 'Estimated Monthly Cost' and '\$0.00'.



Une base de données PostgreSQL est alors ajoutée au projet :

The screenshot shows the Heroku Add-ons page. A single add-on, "Heroku Postgres", is listed. It is attached to the "DATABASE" of the "ocpizza-api" application. The plan is "Hobby Dev" and it is "Free". The estimated monthly cost is \$0.00. A tooltip indicates that the app is currently set to be a "DATABASE". There is also a link to "Manage Attachments".

L'URL de cette base de données est indiquée dans l'onglet **Settings** de l'application et une variable d'environnement est créée pour cette URL, elle s'appelle **DATABASE\_URL** :

The screenshot shows the Heroku App Settings page for the "ocpizza-api" application. The "Settings" tab is selected. In the "Config Vars" section, there is one entry: "DATABASE\_URL" with the value "postgres://ocyjsqbzxyleex:...". Other configuration details like Region (Europe), Stack (heroku-20), and Framework (No framework detected) are also visible.

Il convient maintenant d'utiliser cette variable d'environnement dans le projet

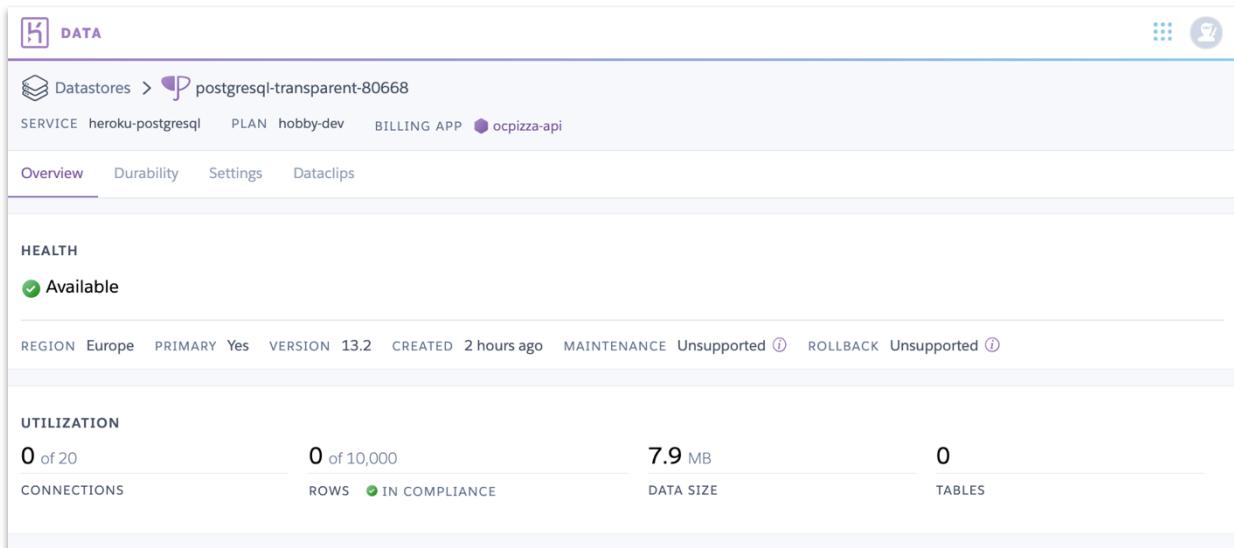
#### 4.1.1.2 - Crédation des tables SQL dans la base de données d'Heroku

Cliquer sur **Heroku Postgres** dans l'onglet **Resources** du tableau de bord d'Heroku :



The screenshot shows the Heroku Resources dashboard. A card for the "Heroku Postgres" service is visible, indicating it is "Attached as DATABASE" and has a "Hobby Dev" plan with "Free" usage. There is also a dropdown arrow icon.

Le tableau de bord de la base de données s'ouvre alors dans un nouvel onglet du navigateur :



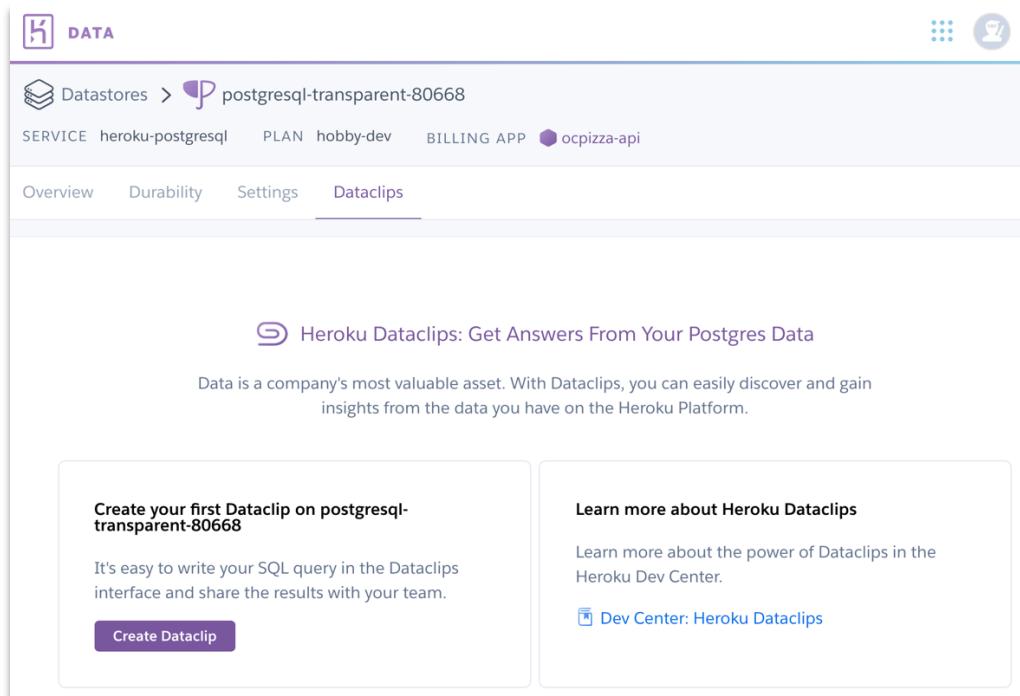
The screenshot shows the Heroku Data dashboard for the "postgresql-transparent-80668" database. It includes sections for Overview, Durability, Settings, and Dataclips. The Overview section displays the following details:

- HEALTH:** Available
- REGION:** Europe
- PRIMARY:** Yes
- VERSION:** 13.2
- CREATED:** 2 hours ago
- MAINTENANCE:** Unsupported ⓘ
- ROLLBACK:** Unsupported ⓘ

**UTILIZATION:**

0 of 20	0 of 10,000	7.9 MB	0
CONNECTIONS	ROWS IN COMPLIANCE	DATA SIZE	TABLES

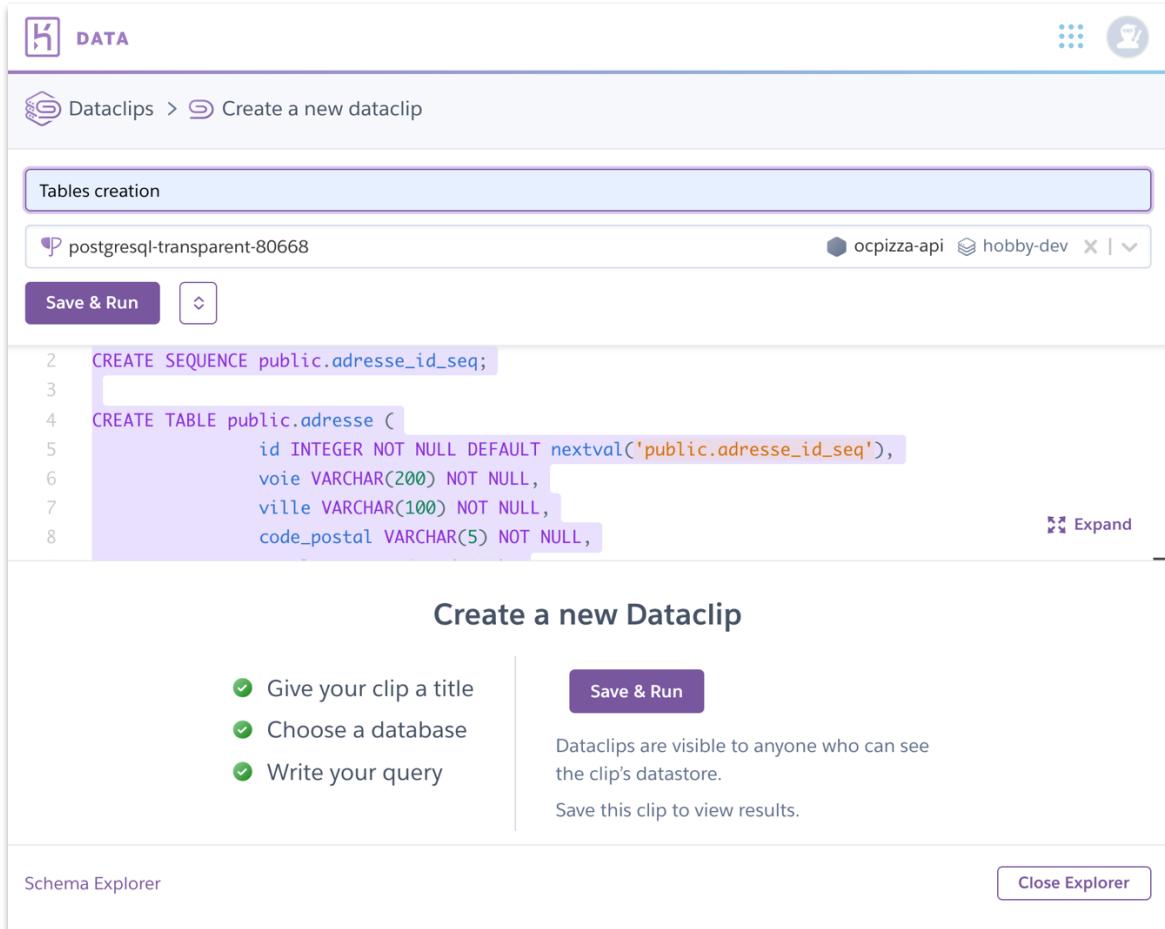
Se rendre dans l'onglet **Dataclips** de ce tableau de bord et cliquer sur le bouton **Create Dataclip** :



The screenshot shows the Heroku Dataclips interface. The "Dataclips" tab is selected. A prominent "Create Dataclip" button is located at the bottom left. The interface includes a heading "Heroku Dataclips: Get Answers From Your Postgres Data" and a brief description: "Data is a company's most valuable asset. With Dataclips, you can easily discover and gain insights from the data you have on the Heroku Platform." Below this are two call-to-action boxes:

- Create your first Dataclip on postgresql-transparent-80668**: "It's easy to write your SQL query in the Dataclips interface and share the results with your team." with a "Create Dataclip" button.
- Learn more about Heroku Dataclips**: "Learn more about the power of Dataclips in the Heroku Dev Center." with a link "Dev Center: Heroku Dataclips".

Rentrer la requête de création des tables dans l'interface proposée et valider :



The screenshot shows the Dataclips interface with the following details:

- Header:** DATA, a user icon, and a grid icon.
- Breadcrumbs:** Dataclips > Create a new dataclip.
- Section:** Tables creation.
- Database:** postgresql-transparent-80668 (selected), ocpizza-api, hobby-dev.
- Buttons:** Save & Run, a dropdown arrow.
- Code Area:**

```

2 CREATE SEQUENCE public.adresse_id_seq;
3
4 CREATE TABLE public.adresse (
5     id INTEGER NOT NULL DEFAULT nextval('public.adresse_id_seq'),
6     voie VARCHAR(200) NOT NULL,
7     ville VARCHAR(100) NOT NULL,
8     code_postal VARCHAR(5) NOT NULL,

```

A "Expand" button is visible to the right of the code area.
- Modal: Create a new Dataclip**
  - Left Column:** A list of steps with checkmarks: Give your clip a title, Choose a database, Write your query.
  - Right Column:** Buttons for Save & Run, Close Explorer, and instructions: Dataclips are visible to anyone who can see the clip's datastore, Save this clip to view results.
- Bottom Buttons:** Schema Explorer, Close Explorer.

La requête de création des tables de la base de données est disponible dans le dépôt GitHub de la documentation du projet :

[https://github.com/centvingt/OCPizzaRedaction/blob/main/db/sql/1\\_create-db-ocpizza.sql](https://github.com/centvingt/OCPizzaRedaction/blob/main/db/sql/1_create-db-ocpizza.sql)

#### **4.1.1.3 - Sauvegarde de la base de données d'Heroku**

Le tableau de bord de la base de données permet d'effectuer des sauvegardes manuelles de la base de données dans l'onglet **Durability** :

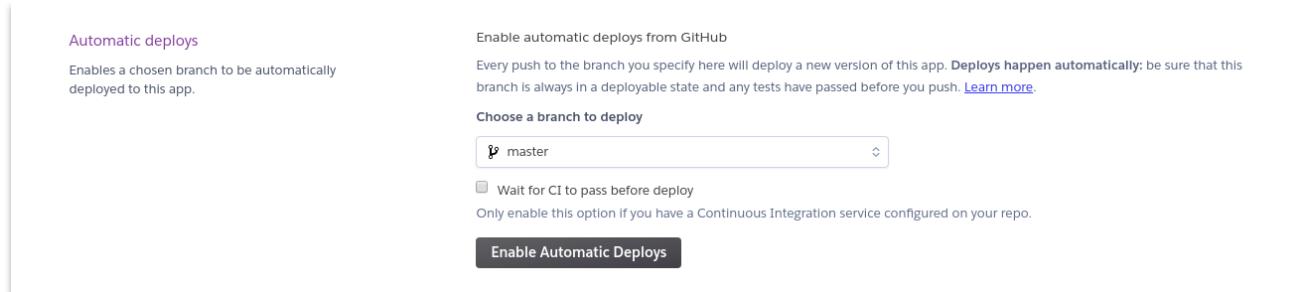
#### Manual Backups & Data Exports

[Learn more](#) about creating exportable copies of your data to use for testing or development, or as additional backups, with pg-backups.

[Create Manual Backup](#)

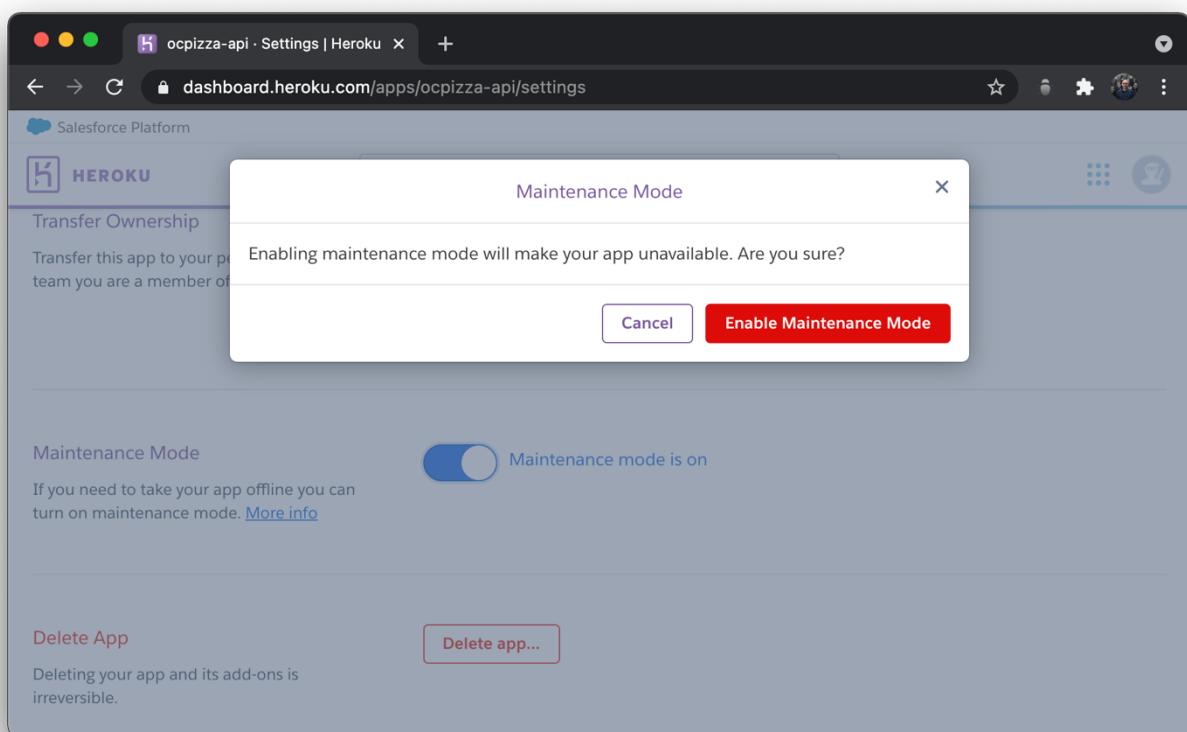
#### 4.1.1.4 - Déploiement continu de l'API sur Heroku

Pour que déploiement des modifications de l'API se fasse automatiquement à chaque modification du dépôt GitHub d'origine, rendez-vous dans la partie **Automatic Deployes** et cliquez sur le bouton **Enable Automatic Deployes** :



#### 4.1.2 - Stopper l'API sur Heroku

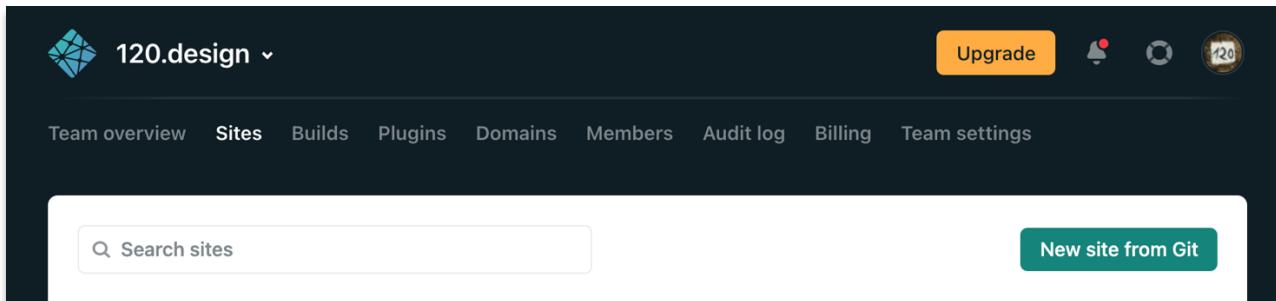
Pour stopper le processus de l'API, se rendre dans la partie l'onglet **Settings** du tableau de bord et activer le mode de maintenance :



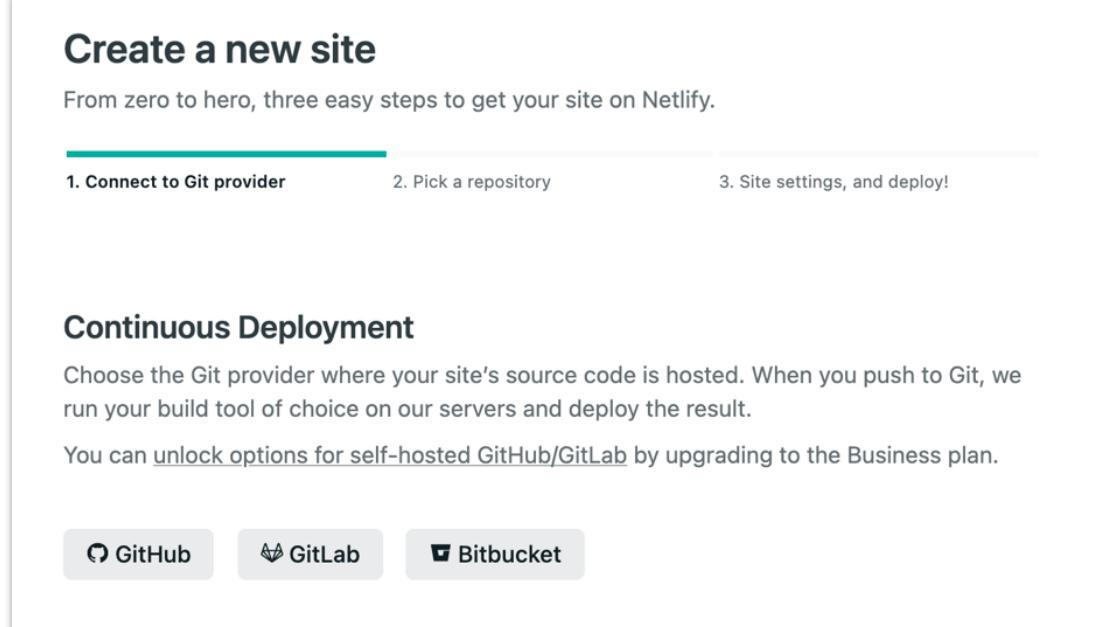
## 4.2 - Déploiement du site statique sur Netlify

### 4.2.1 - Ajout du site statique au compte Netlify

Se connecter au tableau de bord du compte Netlify et cliquer sur le bouton **New site from Git** de l'onglet Sites :



Cliquer sur le bouton **GitHub** :



**Create a new site**  
From zero to hero, three easy steps to get your site on Netlify.

1. Connect to Git provider      2. Pick a repository      3. Site settings, and deploy!

**Continuous Deployment**  
Choose the Git provider where your site's source code is hosted. When you push to Git, we run your build tool of choice on our servers and deploy the result.  
You can unlock options for self-hosted GitHub/GitLab by upgrading to the Business plan.

**GitHub**    **GitLab**    **Bitbucket**



Sélectionner le dépôt GitHub du site statique

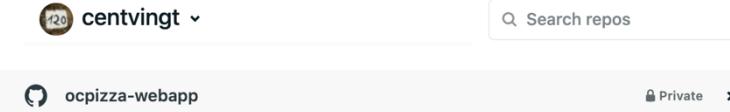
**Create a new site**

From zero to hero, three easy steps to get your site on Netlify.

1. Connect to Git provider    2. Pick a repository    3. Site settings, and deploy!

**Continuous Deployment: GitHub App**

Choose the repository you want to link to your site on Netlify. When you push to Git, we run your build tool of choice on our servers and deploy the result.



Can't see your repo here? [Configure the Netlify app on GitHub](#).

Définir la commande qui lancera l'application comme étant **gatsby build**, puis cliquer sur **Deploy site** :

**Create a new site**

From zero to hero, three easy steps to get your site on Netlify.

1. Connect to Git provider    2. Pick a repository    3. Site settings, and deploy!

**Site settings for centvingt/ocpizza-webapp**

Get more control over how Netlify builds and deploys your site with these settings.

Owner: 120.design

Branch to deploy: main

**Basic build settings**

If you're using a static site generator or build tool, we'll need these settings to build your site. [Learn more in the docs](#)

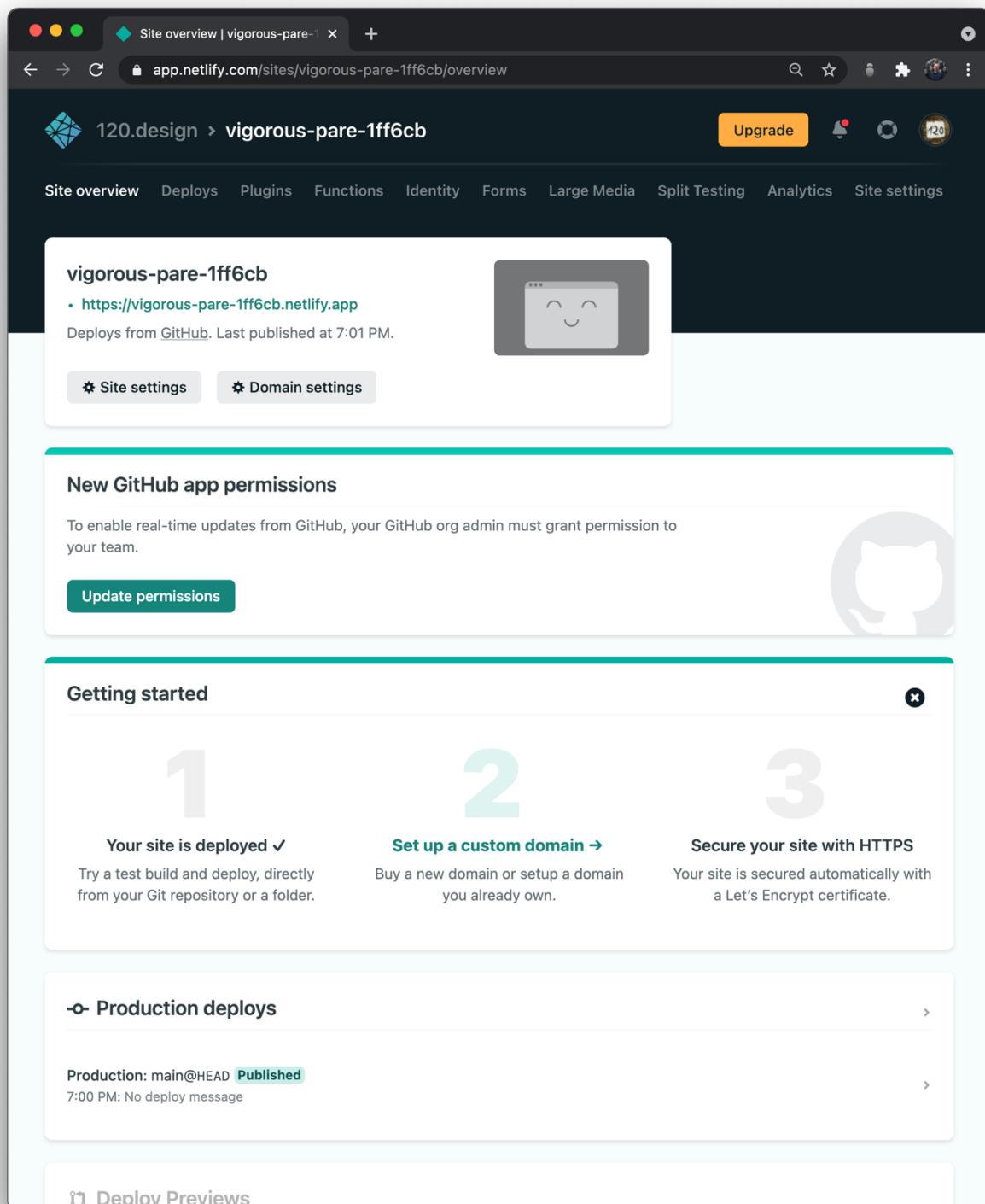
Build command: gatsby build

Publish directory: public

Show advanced

**Deploy site**

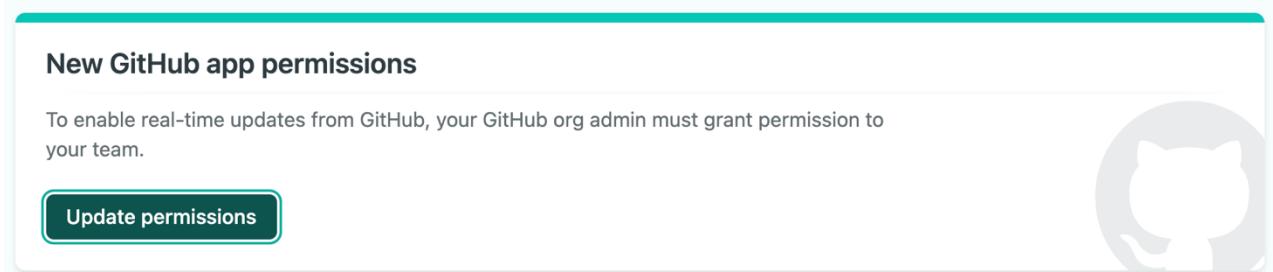
Une fois le site déployé, une URL aléatoire lui est attribué qu'il convient de remplacer par un nom de domaine :



The screenshot shows the Netlify Site overview page for the 'vigorous-pare-1ff6cb' site. At the top, it displays the site's name, a preview icon showing a simple smiley face, and navigation links for Site overview, Deployments, Plugins, Functions, Identity, Forms, Large Media, Split Testing, Analytics, and Site settings. Below this, a card for the site 'vigorous-pare-1ff6cb' shows its URL (<https://vigorous-pare-1ff6cb.netlify.app>), deployment status from GitHub, and a 'Deploy from GitHub' button. Two buttons for Site settings and Domain settings are visible. A large green banner at the bottom of the main content area says 'New GitHub app permissions' and 'To enable real-time updates from GitHub, your GitHub org admin must grant permission to your team.' It includes a 'Update permissions' button and the GitHub logo. The 'Getting started' section is divided into three numbered steps: 1. Your site is deployed (with a checkmark), 2. Set up a custom domain (with a right-pointing arrow), and 3. Secure your site with HTTPS. Step 1 has a note about trying a test build. Step 2 has a note about buying a new domain or using an existing one. Step 3 notes automatic SSL security. Below these are sections for 'Production deploys' (showing a main branch deployment as 'Published' at 7:00 PM) and 'Deploy Previews'. The bottom of the page features a footer with the company name {{Entreprise}}D Itcd.fr and address information: 168, bd. de Magenta – 75010 Paris – 01 22 33 44 55 66 – itcd.fr S.A.R.L. au capital de 100 000,00 € enregistrée au RCS de Paris SIREN 999 999 999 – Code APE : 6202A.

## 4.2.2 - Déploiement continu du site statique sur Netlify

Pour permettre le déploiement continu du site statique, cliquer sur **Update permissions** :

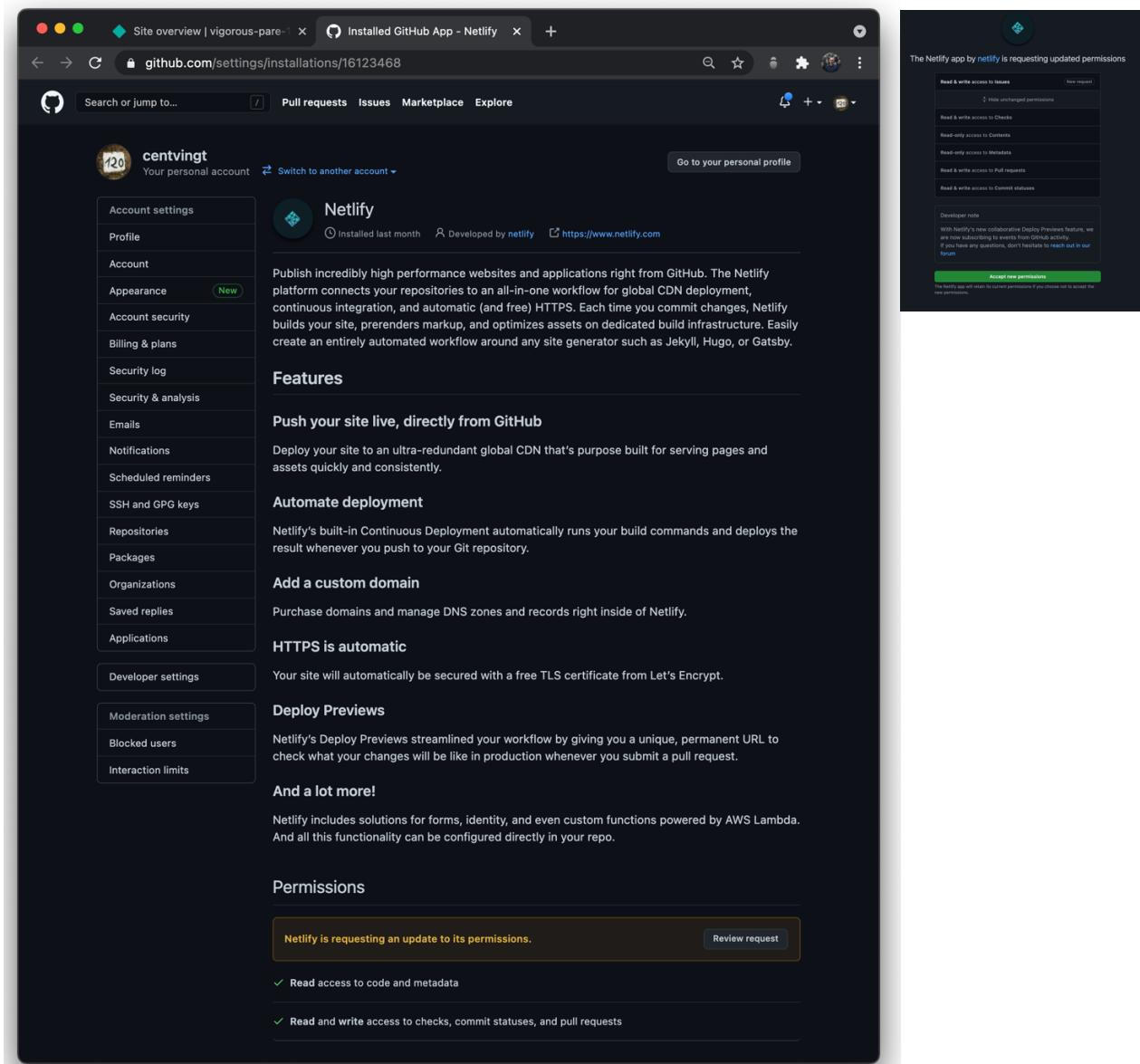


New GitHub app permissions

To enable real-time updates from GitHub, your GitHub org admin must grant permission to your team.

**Update permissions**

On est alors redirigé vers **GitHub** où on peut allouer les autorisations nécessaires à Netlify :



Site overview | vigorous-pare-1 > Installed GitHub App - Netlify

github.com/settings/installations/16123468

centvingt Your personal account Switch to another account Go to your personal profile

**Netlify** Installed last month Developed by netlify https://www.netlify.com

Publish incredibly high performance websites and applications right from GitHub. The Netlify platform connects your repositories to an all-in-one workflow for global CDN deployment, continuous integration, and automatic (and free) HTTPS. Each time you commit changes, Netlify builds your site, prerenders markup, and optimizes assets on dedicated build infrastructure. Easily create an entirely automated workflow around any site generator such as Jekyll, Hugo, or Gatsby.

**Features**

**Push your site live, directly from GitHub**  
Deploy your site to an ultra-redundant global CDN that's purpose built for serving pages and assets quickly and consistently.

**Automate deployment**  
Netlify's built-in Continuous Deployment automatically runs your build commands and deploys the result whenever you push to your Git repository.

**Add a custom domain**  
Purchase domains and manage DNS zones and records right inside of Netlify.

**HTTPS is automatic**  
Your site will automatically be secured with a free TLS certificate from Let's Encrypt.

**Deploy Previews**  
Netlify's Deploy Previews streamlined your workflow by giving you a unique, permanent URL to check what your changes will be like in production whenever you submit a pull request.

**And a lot more!**  
Netlify includes solutions for forms, identity, and even custom functions powered by AWS Lambda. And all this functionality can be configured directly in your repo.

**Permissions**

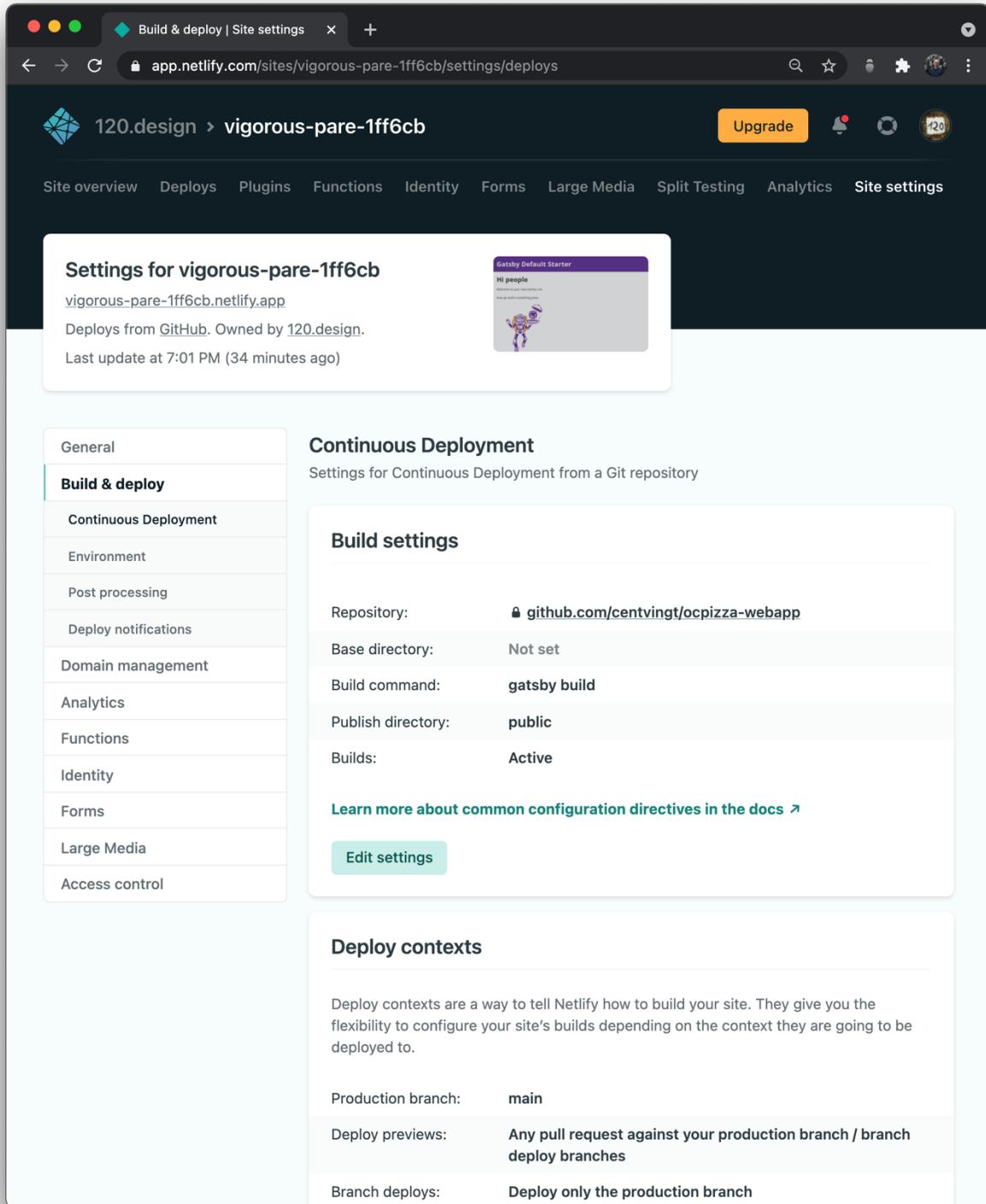
Netlify is requesting an update to its permissions. Review request

✓ Read access to code and metadata

✓ Read and write access to checks, commit statuses, and pull requests

#### 4.2.3 - Stopper le site statique sur Netlify

Pour stopper la distribution du site statique par Netlify, cliquer sur le bouton **Edit settings** de la partie **Build settings** de l'onglet **Build & deploy** de la page **Site settings** :



The screenshot shows the Netlify Site settings page for the site `vigorous-pare-1ff6cb`. The URL in the browser is `app.netlify.com/sites/vigorous-pare-1ff6cb/settings/deploys`.

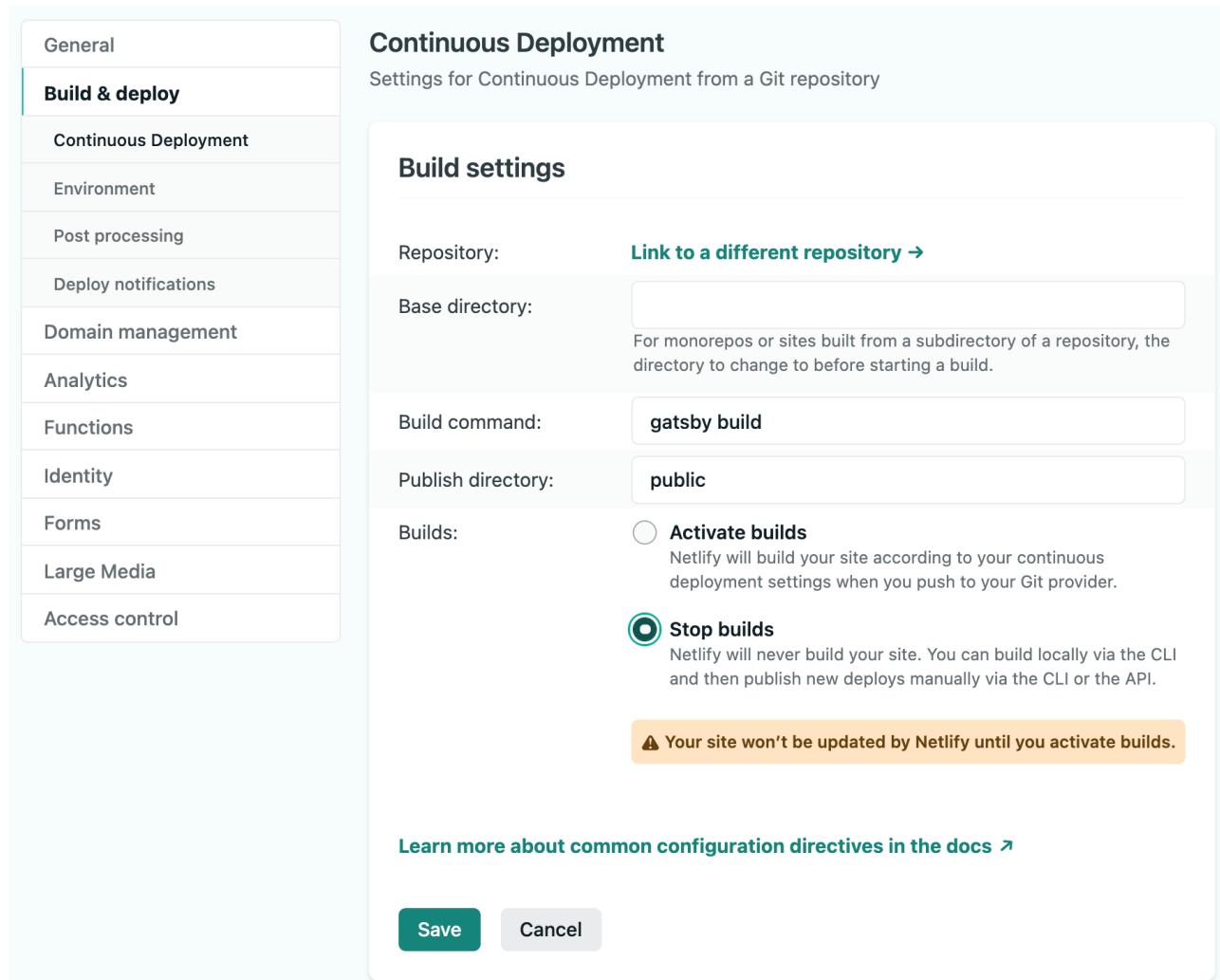
The left sidebar menu is open, showing various sections: General, Build & deploy (which is selected), Continuous Deployment, Environment, Post processing, Deploy notifications, Domain management, Analytics, Functions, Identity, Forms, Large Media, and Access control.

The main content area is titled "Settings for vigorous-pare-1ff6cb" and shows the site's URL as `vigorous-pare-1ff6cb.netlify.app`. It indicates that the site deploys from GitHub and is owned by `120.design`. The last update was at 7:01 PM (34 minutes ago).

The "Build & deploy" section contains the "Continuous Deployment" configuration. Under "Build settings", the repository is set to `github.com/centvingt/ocpizza-webapp`, the base directory is "Not set", the build command is `gatsby build`, the publish directory is `public`, and builds are set to "Active". A link "Learn more about common configuration directives in the docs" is provided, along with a "Edit settings" button.

The "Deploy contexts" section explains that deploy contexts tell Netlify how to build the site. It shows the production branch as `main`, deploy previews as "Any pull request against your production branch / branch deploy branches", and branch deploys as "Deploy only the production branch".

Puis sélectionner l'option **Stop builds** et cliquer sur le bouton **Save** :



**General**

**Build & deploy** (selected)

Continuous Deployment

Environment

Post processing

Deploy notifications

Domain management

Analytics

Functions

Identity

Forms

Large Media

Access control

**Continuous Deployment**

Settings for Continuous Deployment from a Git repository

**Build settings**

Repository: [Link to a different repository →](#)

Base directory:

For monorepos or sites built from a subdirectory of a repository, the directory to change to before starting a build.

Build command: `gatsby build`

Publish directory: `public`

Builds:

**Activate builds**  
Netlify will build your site according to your continuous deployment settings when you push to your Git provider.

**Stop builds**  
Netlify will never build your site. You can build locally via the CLI and then publish new deploys manually via the CLI or the API.

⚠ Your site won't be updated by Netlify until you activate builds.

[Learn more about common configuration directives in the docs ↗](#)

**Save**   **Cancel**

## 4.3 - Publication des application mobiles sur les stores

Les procédures détaillées ici sont les mêmes pour les trois applications mobiles :

1. Celle du client de la pizzéria (OC Pizza)
2. Celle du pizaïolo (OC Pizza Commandes)
3. Celle du livreur (OC Pizza Livraisons)

### 4.3.1 - Configurer app.json

#### 4.3.1.1 - Configurer app.json pour l'application du client de la Pizzéria

Configurer ainsi le fichier **app.json** d'OC Pizza, l'application mobile du client de la pizzéria :

```
{  
  "expo": {  
    "name": "OC Pizza",  
    "icon": "./assets/app-icon.png",  
    "version": "1.0.0",  
    "slug": "oc-pizza",  
    "ios": {  
      "bundleIdentifier": "com.ocpizza.OCPizza",  
      "buildNumber": "1.0.0"  
    },  
    "android": {  
      "package": "com.ocpizza.OCPizza",  
      "versionCode": 1  
    }  
  }  
}
```

Ici, nous avons indiqué le nom, le *slug*, le *bundle identifier* et l'identifiant du *package* d'OC Pizza, l'application du client de la pizzéria. Il convient de modifier ces informations pour les deux autres applications.

#### 4.3.1.2 - Configurer app.json pour l'application du pizzaïolo

Configurer ainsi le fichier **app.json** d'OC Pizza, l'application mobile du pizzaïolo :

```
{  
  "expo": {  
    "name": "OC Pizza Commandes",  
    "icon": "./assets/app-icon.png",  
    "version": "1.0.0",  
    "slug": "oc-pizza-commandes",  
    "ios": {  
      "bundleIdentifier": "com.ocpizza.OCPizzaCommandes",  
      "buildNumber": "1.0.0"  
    },  
    "android": {  
      "package": "com.ocpizza.OCPizzaCommandes",  
      "versionCode": 1  
    }  
  }  
}
```

#### 4.3.1.3 - Configurer app.json pour l'application du livreur

Configurer ainsi le fichier **app.json** d'OC Pizza, l'application mobile du livreur :

```
{  
  "expo": {  
    "name": "OC Pizza Livraisons",  
    "icon": "./assets/app-icon.png",  
    "version": "1.0.0",  
    "slug": "oc-pizza-livraisons",  
    "ios": {  
      "bundleIdentifier": "com.ocpizza.OCPizzaLivraisons",  
      "buildNumber": "1.0.0"  
    },  
    "android": {  
      "package": "com.ocpizza.OCPizzaLivraisons",  
      "versionCode": 1  
    }  
  }  
}
```

#### 4.3.2 - Publier les applications mobiles sur l'App Store

Pour chacun des trois projets (l'application mobile pour le client de la pizzéria, celle du pizzaïolo et celle du livreur), lancer cette commande dans le Terminal depuis la racine du projet :

```
$ expo build:ios
```

Expo propose alors de se connecter au compte de développeur de l'App Store :

```
$ expo build:ios
[16:44:37] Checking if current build exists...

[16:44:37] No currently active or previous builds for this project.
[16:44:37]
We need your Apple ID/password to manage certificates, keys
and provisioning profiles from your Apple Developer account.

Note: Expo does not keep your Apple ID or your Apple ID password.

? What's your Apple ID? xxx@yyy.zzz
? Password? [hidden]
✓ Authenticated with Apple Developer Portal successfully!
[16:44:46] You have 4 teams associated with your account
? Which team would you like to use? 3) ABCDEFGHIJ "John Turtle" (Individual)
✓ Ensured App ID exists on Apple Developer Portal!
[16:44:59] We do not have some credentials for you: Apple Distribution Certificate, Apple Push Notifications service key, Apple Provision
? How would you like to upload your credentials? (Use arrow keys)
> Expo handles all credentials, you can still provide overrides
  I will provide all the credentials and files needed, Expo does limited validation
```

#### 4.3.3 - Produire les fichier APK pour le Play Store

Pour chacun des trois projets (l'application mobile pour le client de la pizzéria, celle du pizzaïolo et celle du livreur), lancer cette commande dans le Terminal depuis la racine du projet :

```
$ expo build:android -t apk
```