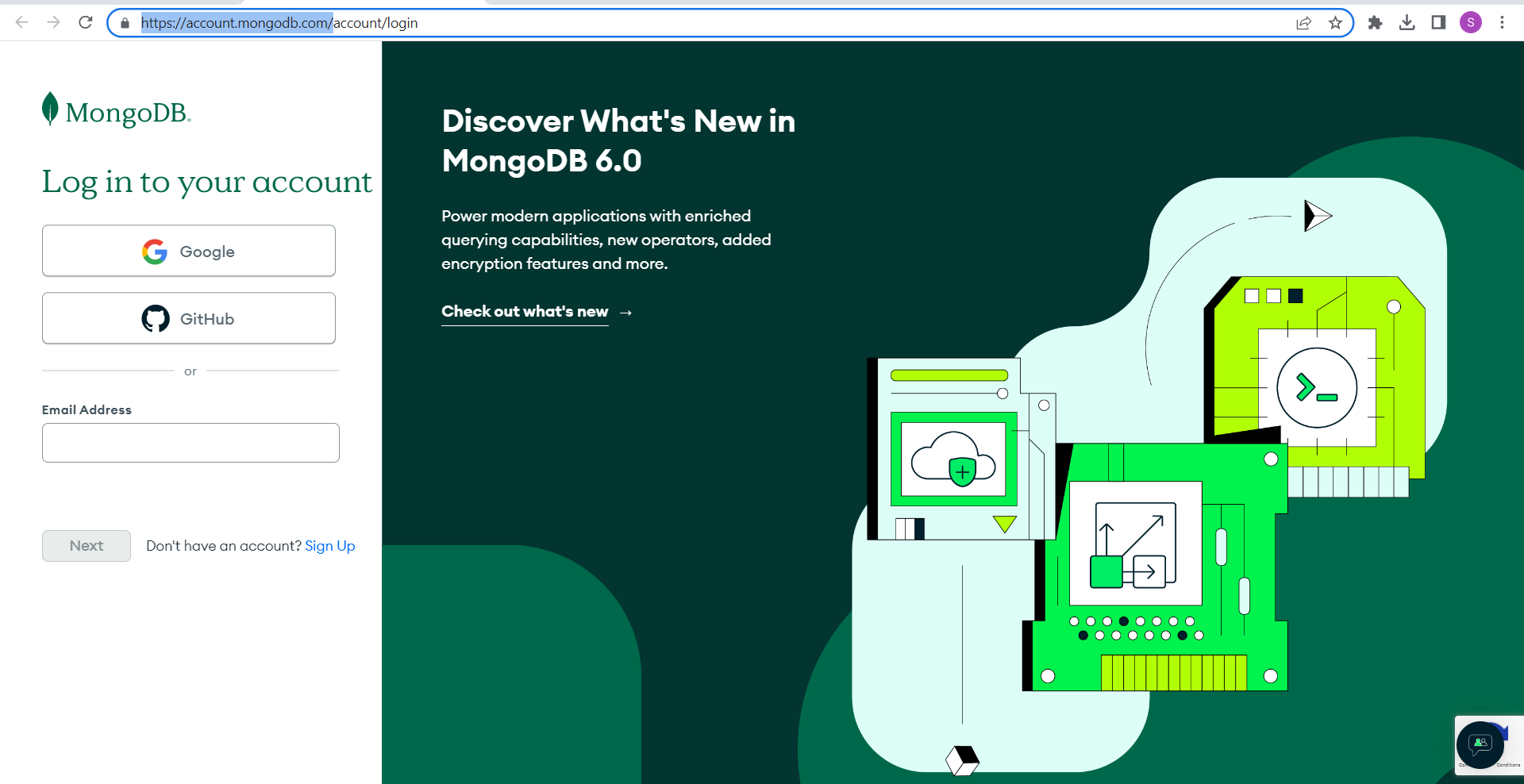
Deploy avec Netlify

Atlas MongoDB

Pour créer une base de données MongoDB avec AtlasUI, ouvrez une fenêtre de navigateur et connectez-vous à : <https://cloud.mongodb.com/>

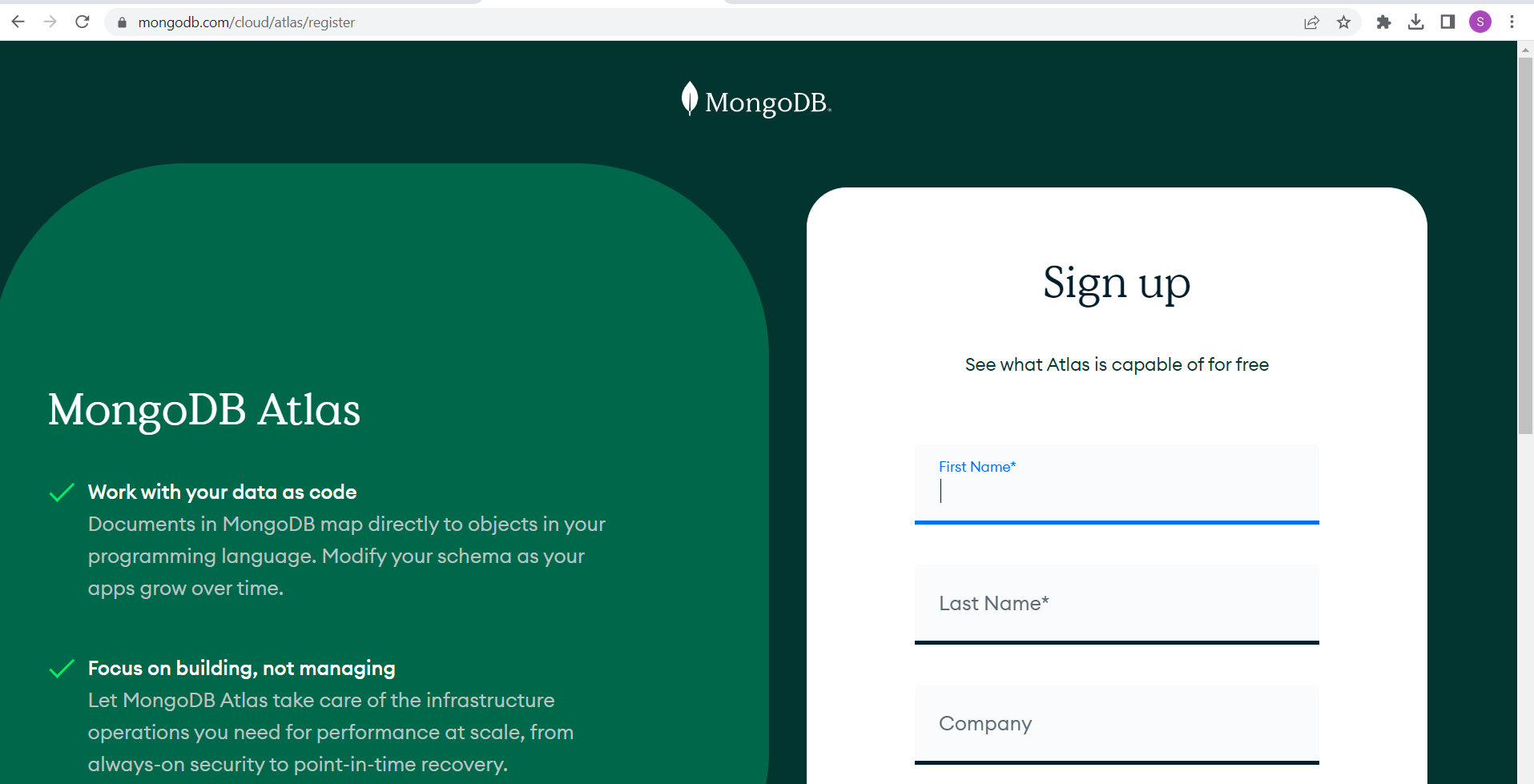


Pour avoir accès au cloud MongoDB et ensuite pouvoir créer votre propre base de données, rendez-vous au <https://www.mongodb.com/atlas>

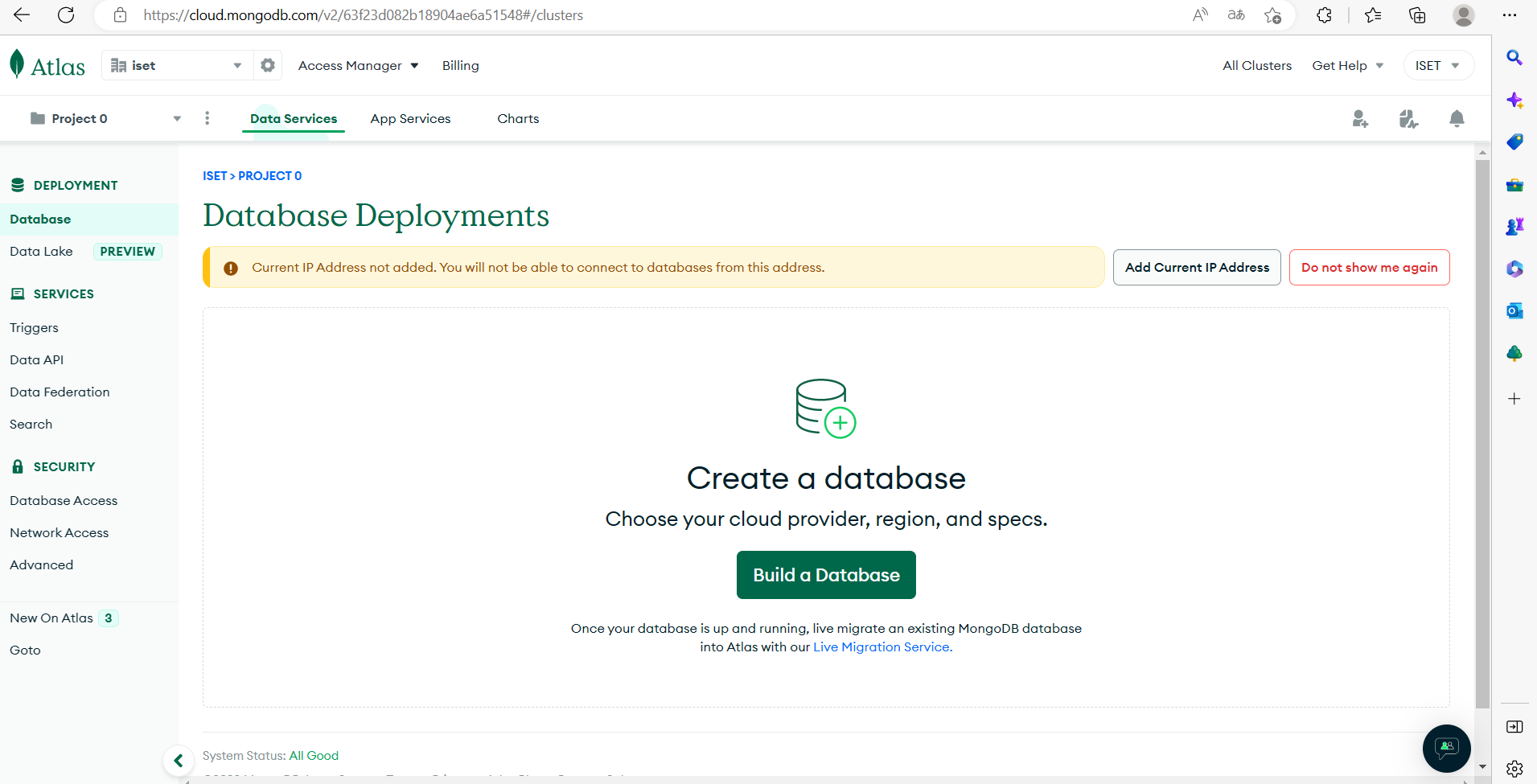
et cliquer sur le bouton "Try Free".



Choisir le service Atlas, remplir le questionnaire.



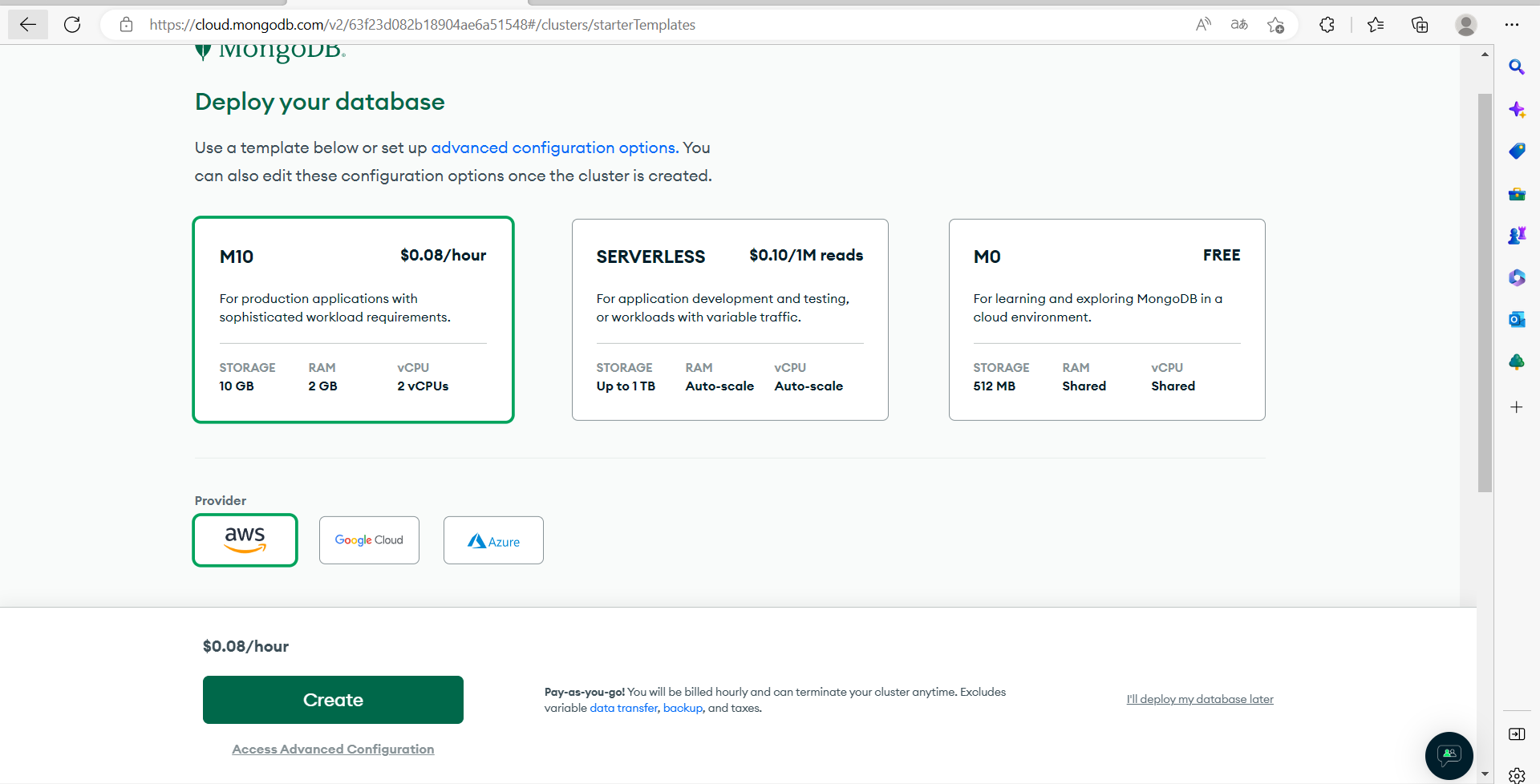
Une fois votre compte créé, rendez-vous sur le panneau de configuration

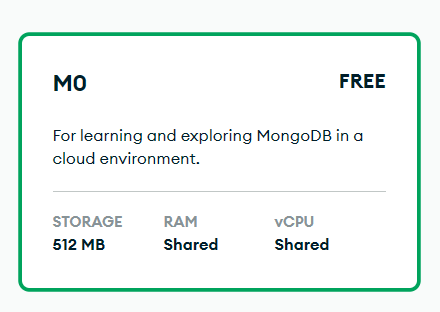


Cliquez sur « Build a Database »

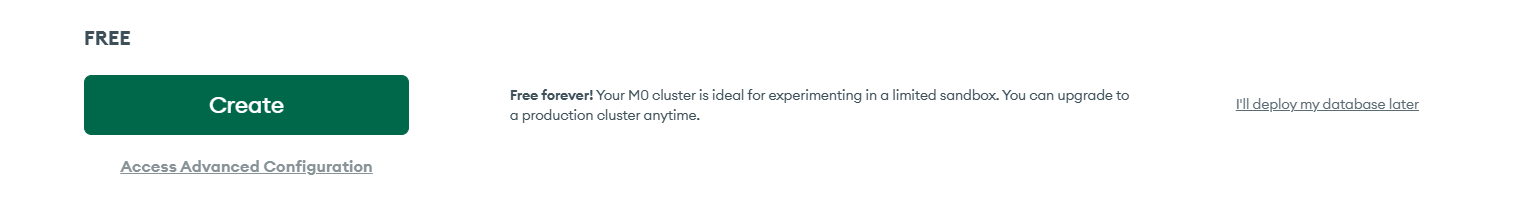


Ensuite choisir l'option «  Free »

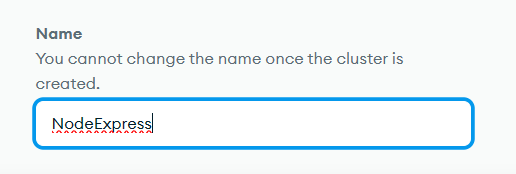




Cliquer ensuite sur « create »

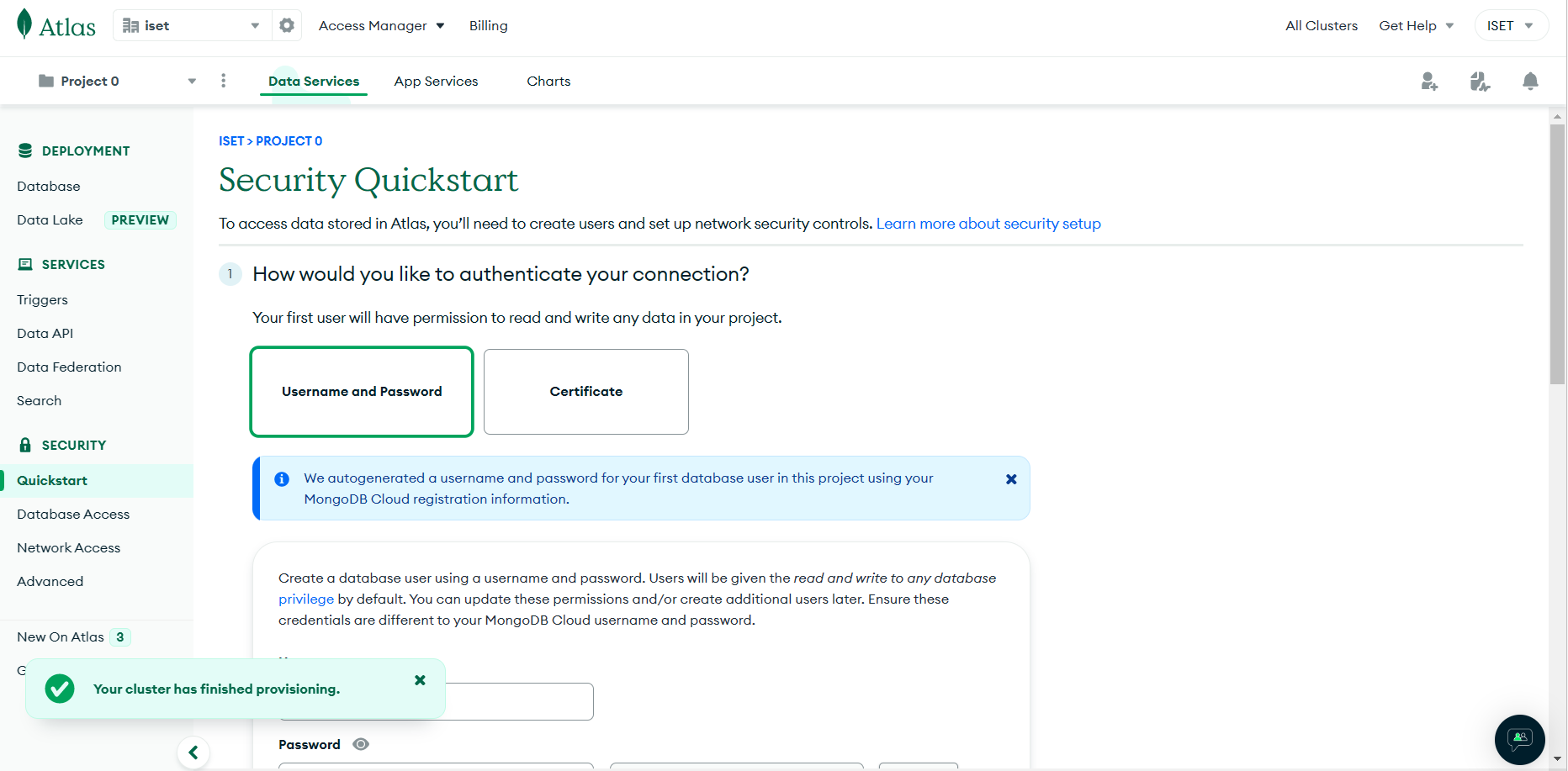


Ensuite, conserver les options par défaut sauf la dernière option qui est "Cluster Name" et changer le nom pour "NodeExpress" par exemple. Enfin cliquer sur "Create".



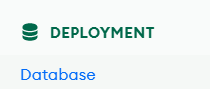
Ce qui affiche

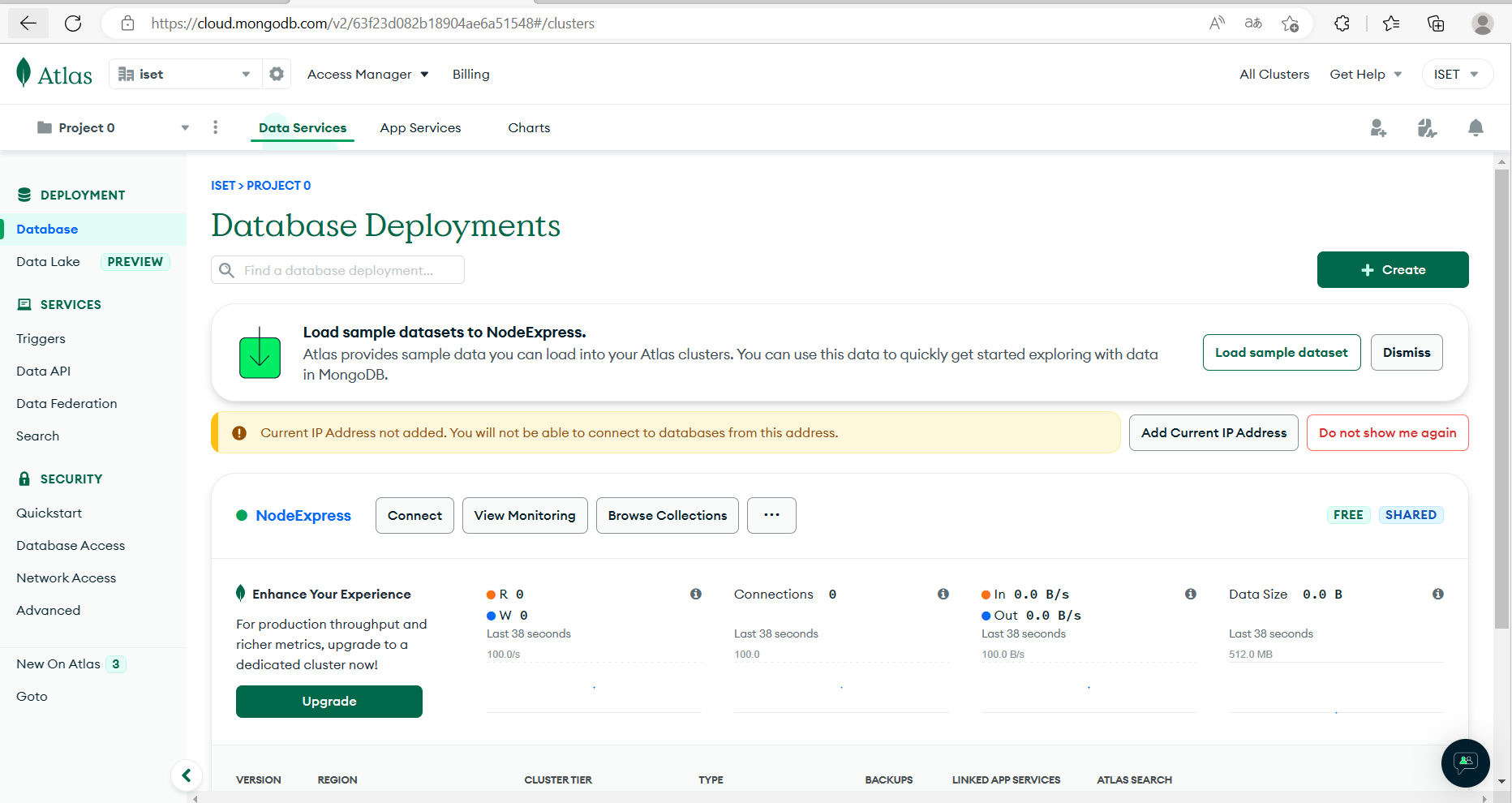




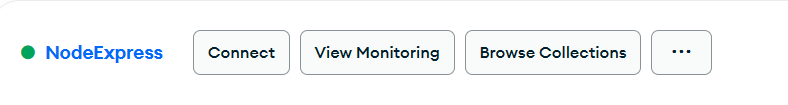
Votre compte MongoDB est créer et activé nous allons maintenant créer notre base de donnée et notre première collection.

Cliquez sur Database

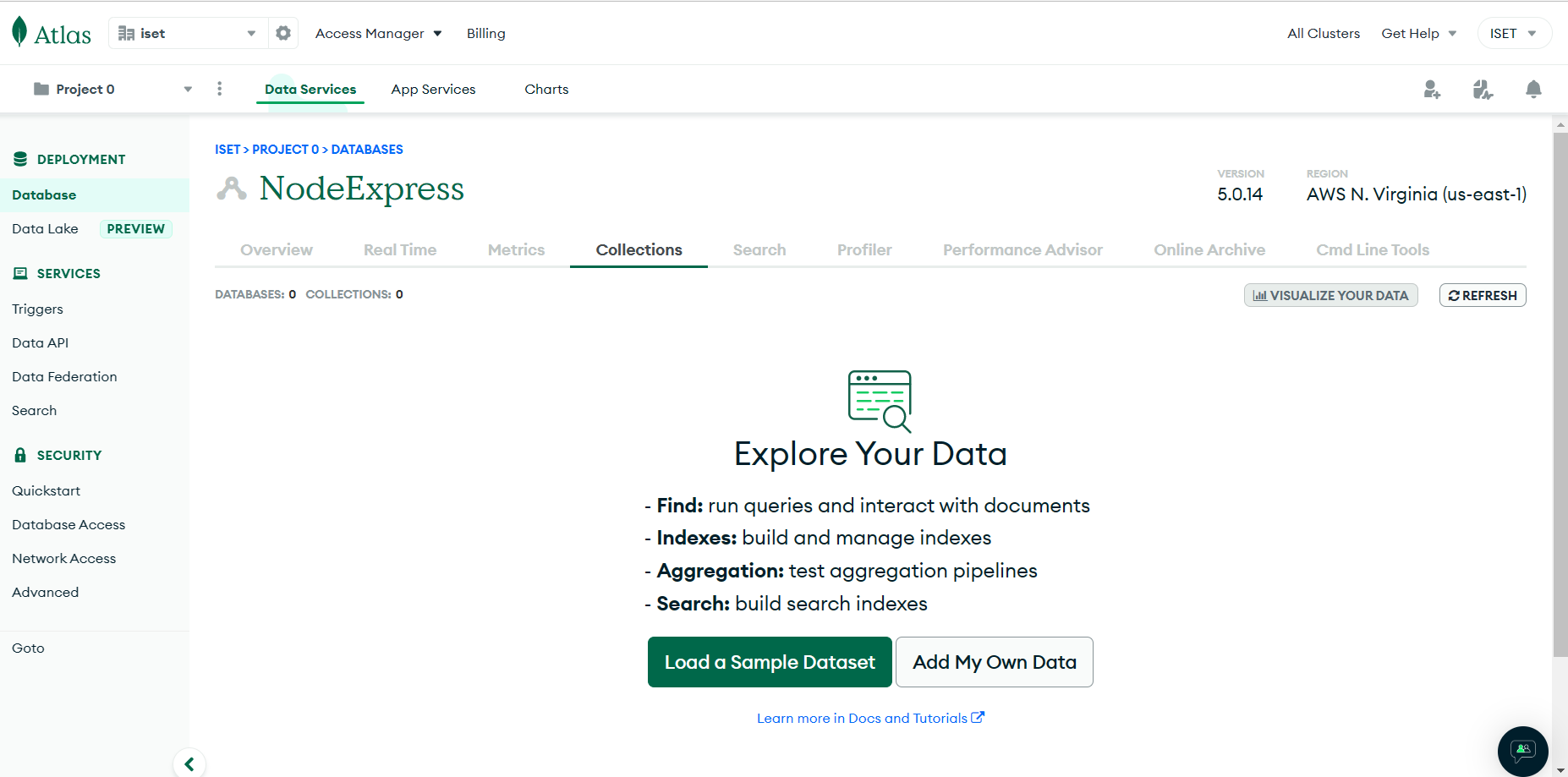




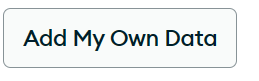
Pour créer une base de données et une collection. Cliquer sur "Browse Collection",



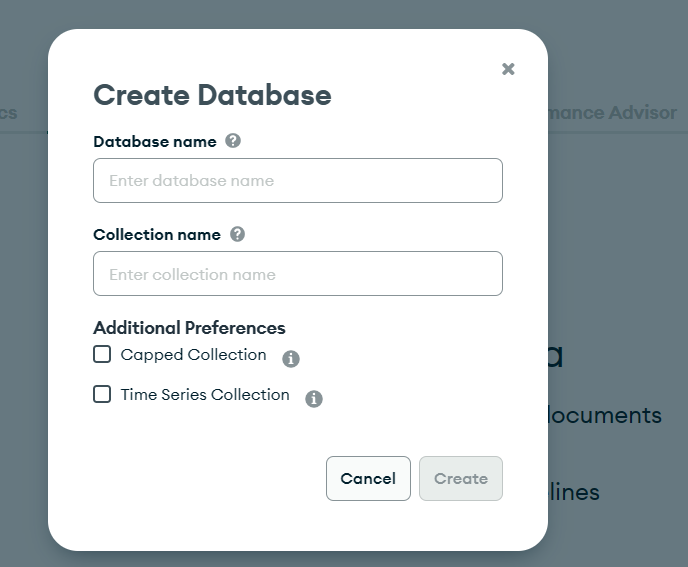
Ayant ce résultat :



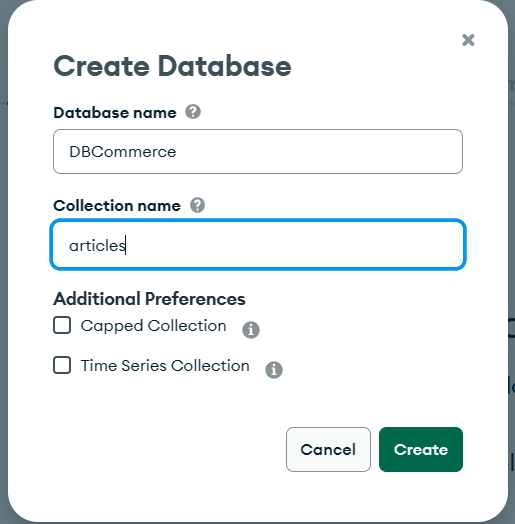
Ensuite sur "Add My Own Data"



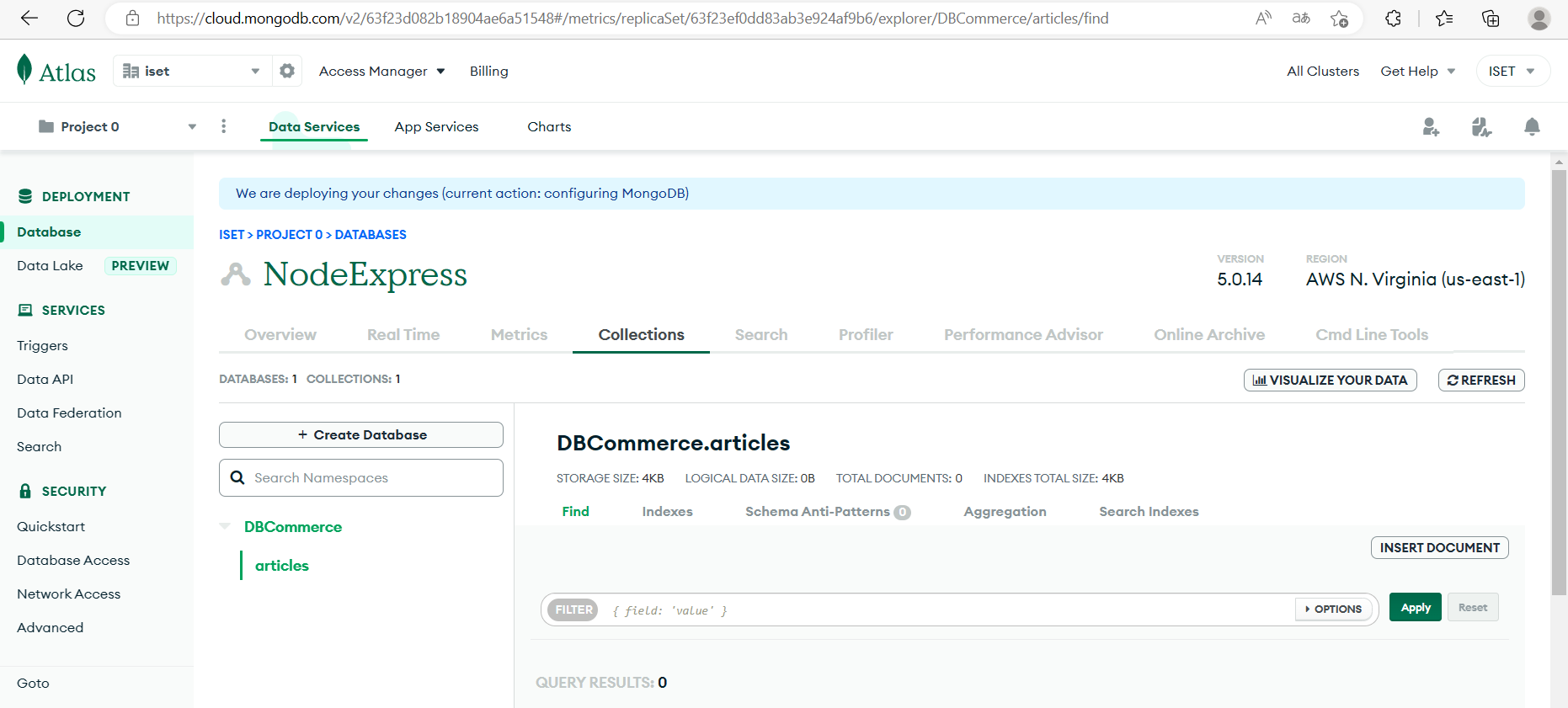
Ce qui affiche cette fenêtre



Enfin saisir le nom de la base de données et le nom de la collection et cliquer sur "Create"



Ce qui donne ce résultat :



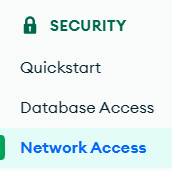
Pour résumer, nous avons créé un compte cloud MongoDB, créer un free Cluster et créer notre première base de donnée et enfin notre première collection.

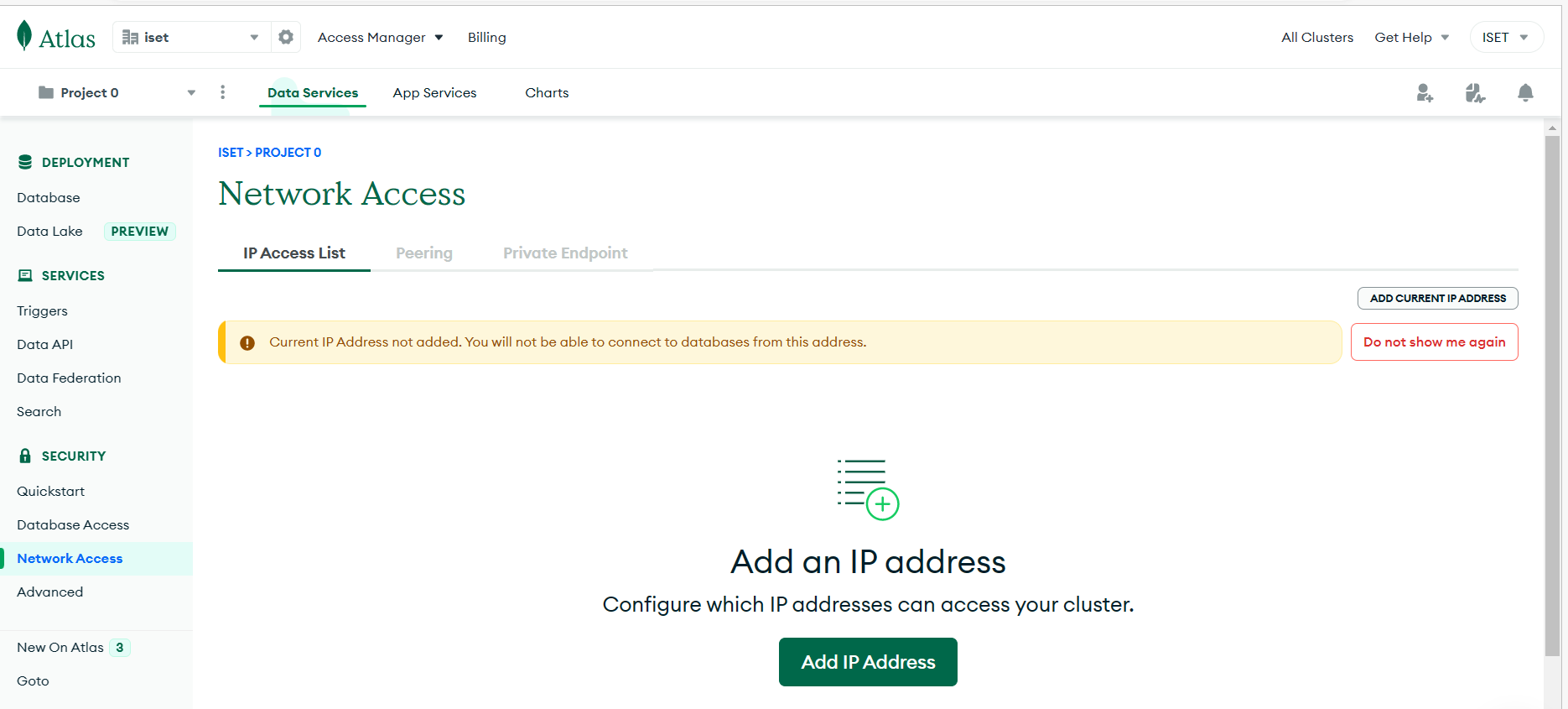
Maintenant nous sommes prêts pour coder notre API et manipuler notre base de données.

Pour connecter à notre base de donnée MongoDB nous avons besoins du URL de connection. MongoDB va nous fournir cet URL.

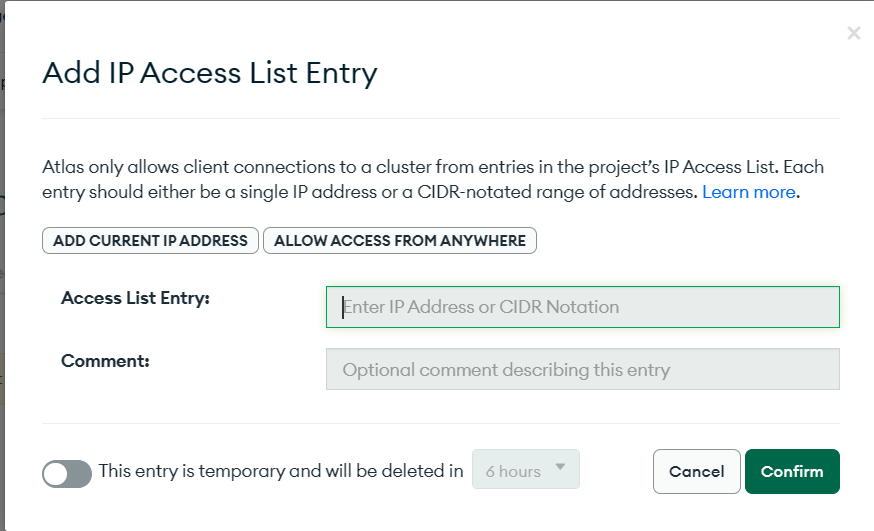
Il faut configurer l’accès des adresses IP qui peuvent accéder à la BD.

Cliquer sur « Network Access »





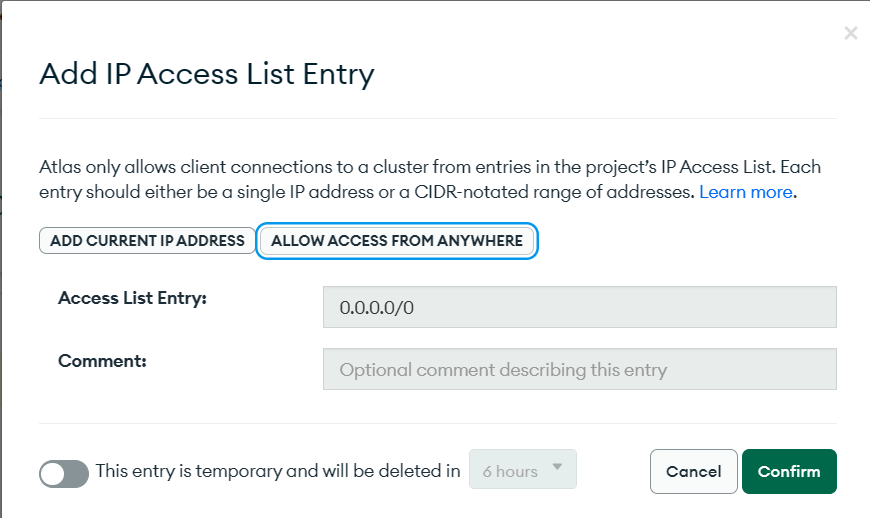
Puis sur « Add IP Adress »



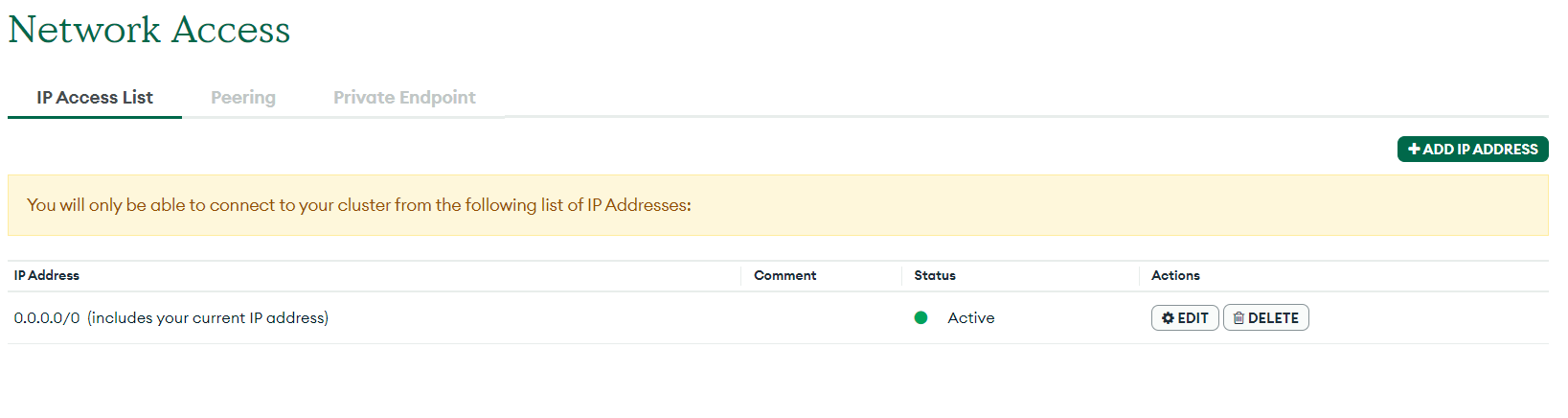
Cliquer sur :



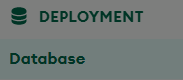
Ce qui affiche



Puis cliquer sur « Confirm » pour avoir ce résultat.



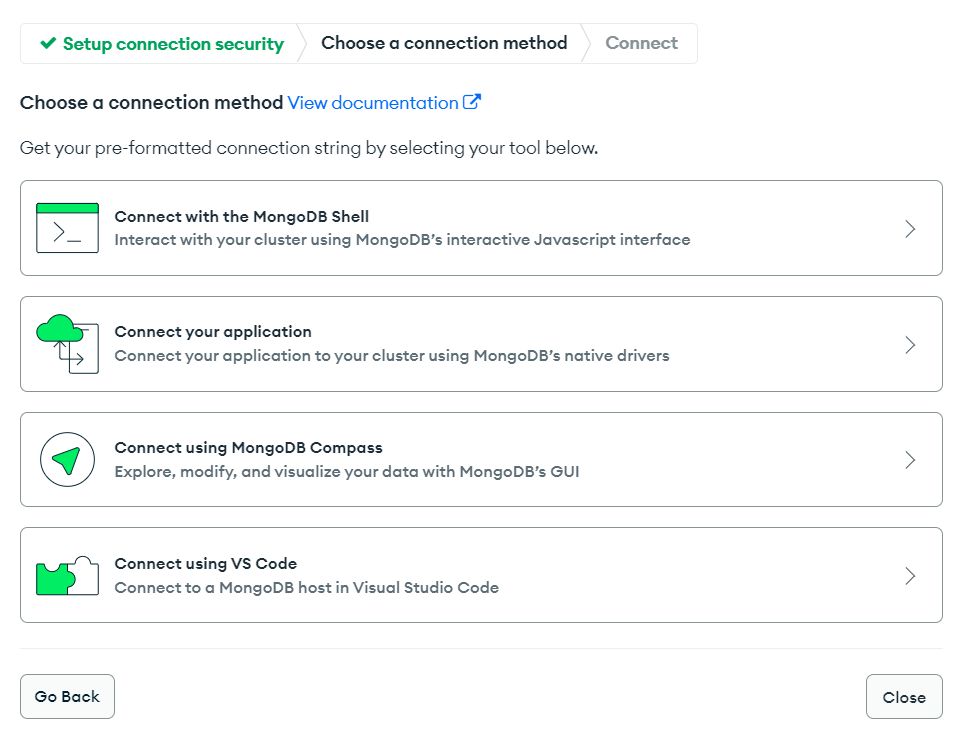
Revenir à la rubrique Database



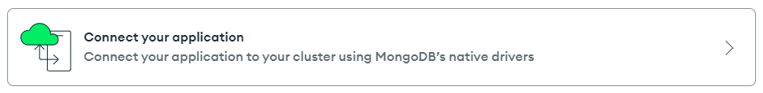
Cliquer sur « Connect »



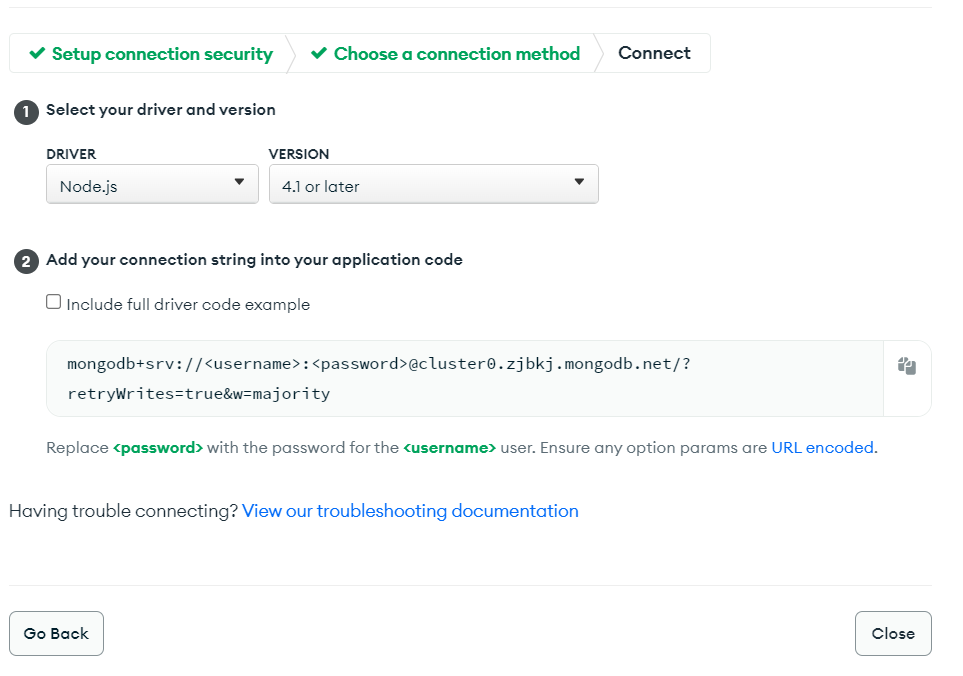
Ce qui donne



Choisir



Ce qui affiche :



Copier la chaîne de connexion qui sera affectée à la variable d’environnement DATABASECLOUD dans l’application Express pour pouvoir faire la connexion. Il faut remplacer les valeurs de username et password.

mongodb+srv://<username>:<password>@cluster0.zjbkj.mongodb.net/?retryWrites=true&w=majority

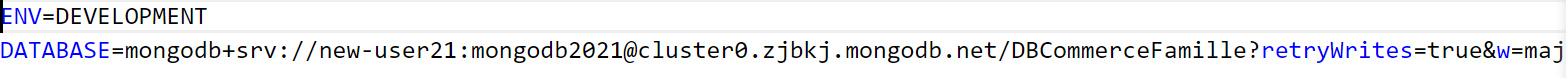
Puis cliquer sur « Close ».

Nous avons donc dans l’application Express un fichier qui contiendra notre URL de connexion. Ce fichier se nomme ".env" (dot env)

Vous pouvez créer ce fichier à la racine du projet s’il n’existe pas et inclure le URL de connexion MongoDB

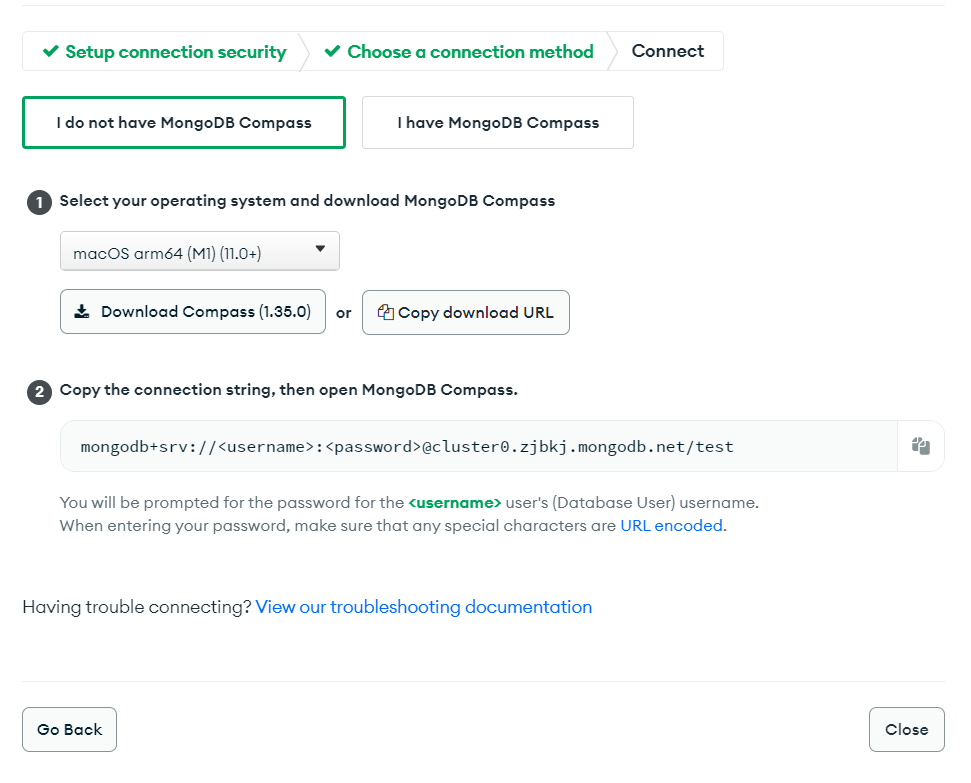
Remplacer et avec votre nom d'utilisateur et mot de passe utilisateur cloud MongoDB et également remplacer  "?" avec le nom de votre base de données

**Exemple  .env :**

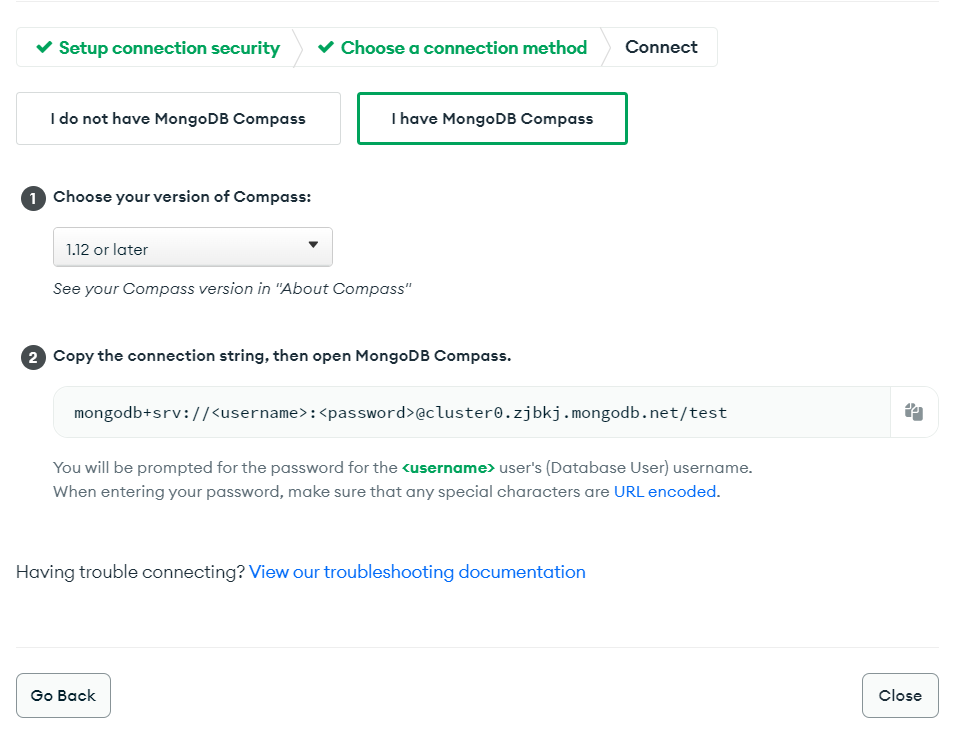


**Remarque** : si on choisit « Connect using mongodb compass »

ça donne



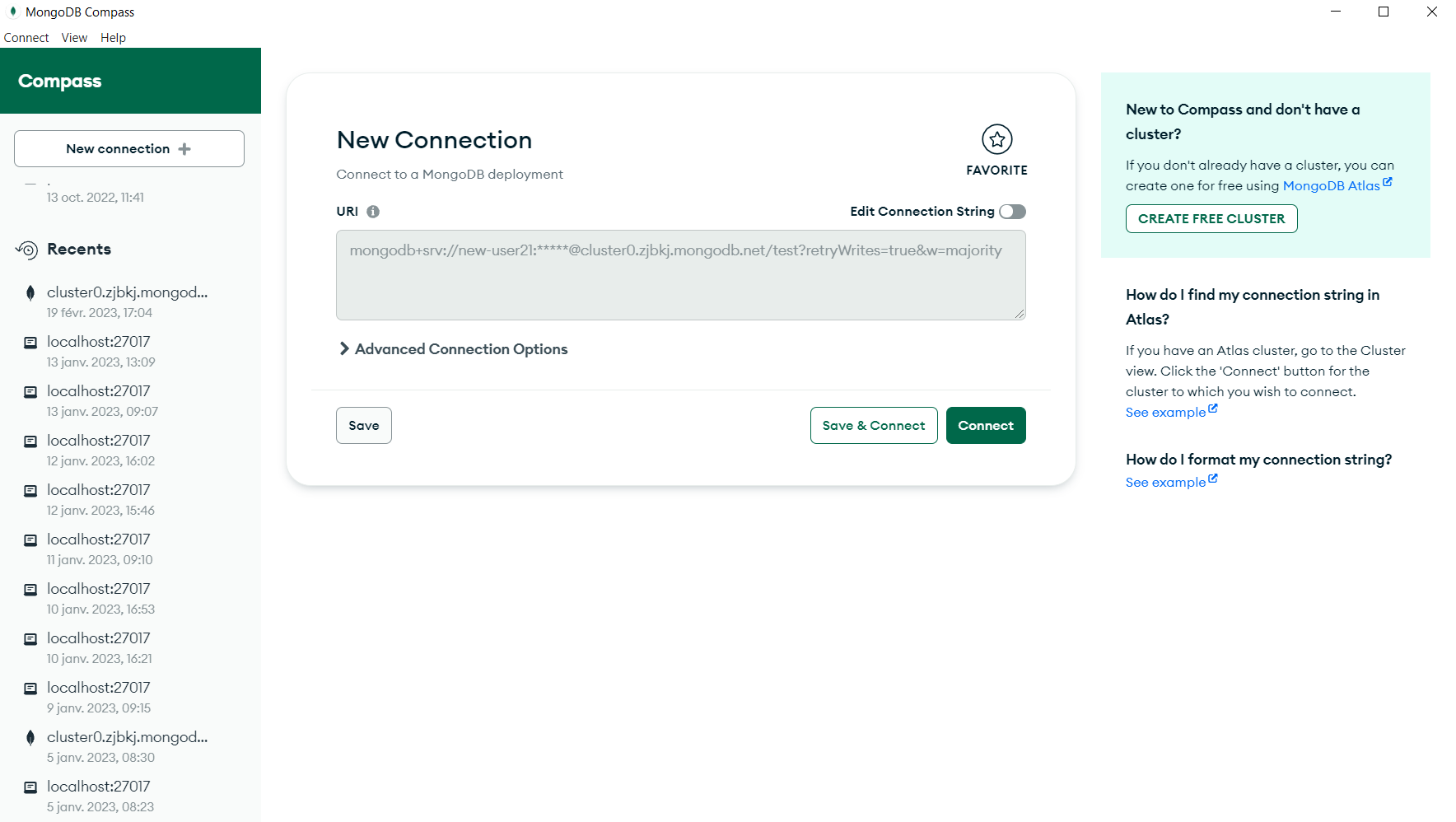
S’il est préalablement créé choisir « I have MongoDB Compass »

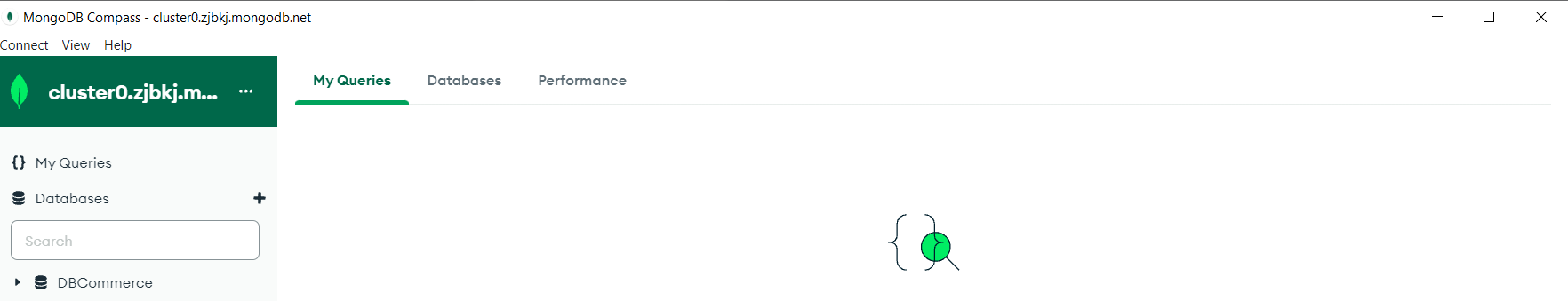


Tester dans mongoDBCompass









Puis dans le fichier .env du projet NextJS cahnger la variable de connexion de la base de données



NEXT\_PUBLIC\_DATABASE\_URL=mongodb+srv://hassan:messk2015@cluster0.yfhhu.mongodb.net/bibliothequedb?retryWrites=true&w=majority

Puis tester l’application



L'approche la plus courante du CORS consiste à le configurer globalement sur tous les points de terminaison de l'API. Il existe de nombreuses façons d’y parvenir dans Next.js.

Vous pouvez définir manuellement les en-têtes CORS dans Next.js grâce à la clé headers dans next.config.js :



/\*\* @type {import('next').NextConfig} \*/

const nextConfig = {

images:{

domains:["res.cloudinary.com","static.vecteezy.com"]

},

env:{

NEXT\_PUBLIC\_API\_URL : "https://fullstacknextjs-rouge.vercel.app/api",

NEXT\_PUBLIC\_GOOGLE\_CLIENTID : "31017781977-c2mev6djp0jd9dte3hcqal7v8ha2tp27.apps.googleusercontent.com",

NEXT\_PUBLIC\_GOOGLE\_CLIENTSECRET : "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",

NEXT\_PUBLIC\_AUTH\_SECRET : "secret",

NEXT\_PUBLIC\_STRIPE\_PUBLISHABLE\_KEY : "pk\_test\_51KtYRUD3HS4vNAwatvmqAEXLKKX11UOcpkHfLnw9UPI9kZ7AJCOeLkqik61wHFXLmRGHUd4aNBvp45v82DpskKl300bMfznwlE",

STRIPE\_SECRET\_KEY : "sk\_test\_51KtYRUD3HS4vNAwa7ANL32HQqRTywhV3JHWIp3BxAIHv04bWoz22aKlRs9Q1L6znSX2i5fu5i3Xkl9i2Goz7jAkC00LL0T3lTL",

NEXT\_PUBLIC\_URL : "https://fullstacknextjs-rouge.vercel.app",

NEXT\_PUBLIC\_DATABASE\_URL : "mongodb+srv://hassan:messk2015@cluster0.yfhhu.mongodb.net/bibliothequedb?retryWrites=true&w=majority"

} ,

async headers() {

return [

{

source: '/:path\*',

headers: [

{ key: 'Access-Control-Allow-Credentials', value: 'true' },

{ key: 'Access-Control-Allow-Origin', value: '\*' },

{ key: 'Access-Control-Allow-Methods', value: 'GET, DELETE, PATCH, POST, PUT' },

{ key: 'Access-Control-Allow-Headers', value: 'X-CSRF-Token, X-Requested-With, Accept, Accept-Version, Content-Length, Content-MD5, Content-Type, Date, X-Api-Version' },

],

},

];

},

}

module.exports = nextConfig

/\*\* @type {import('next').NextConfig} \*/

const nextConfig = {

    async headers() {

        return [

          {

            source: '/:path\*',

            headers: [

              { key: 'Access-Control-Allow-Credentials', value: 'true' },

              { key: 'Access-Control-Allow-Origin', value: '\*' },

              { key: 'Access-Control-Allow-Methods', value: 'GET, DELETE, PATCH, POST, PUT' },

              { key: 'Access-Control-Allow-Headers', value: 'X-CSRF-Token, X-Requested-With, Accept, Accept-Version, Content-Length, Content-MD5, Content-Type, Date, X-Api-Version' },

            ],

          },

        ];

      },

};

export default nextConfig;

Déposer le projet dans Github

On va déposer le projet dans GitHub

Dans le répertoire du projet dans l’invite de commande (cmd) taper :

git init

Puis taper :

git add .

Taper :

git commit -m "mon premier commit"

Aller à la rubrique Repositories dans le compte GitHub

Puis cliquer sur New

Copiez le lien vers le répertoire, situé généralement en haut du projet dans une boite bien visible : le lien se termine par « .git »

Dans l’invite de commande taper :

git remote add origin Le lien copié

Taper par la suite :

git remote -v

pour la vérification que les choses fonctionnent dans les deux sens.

Enfin taper

git push origin master

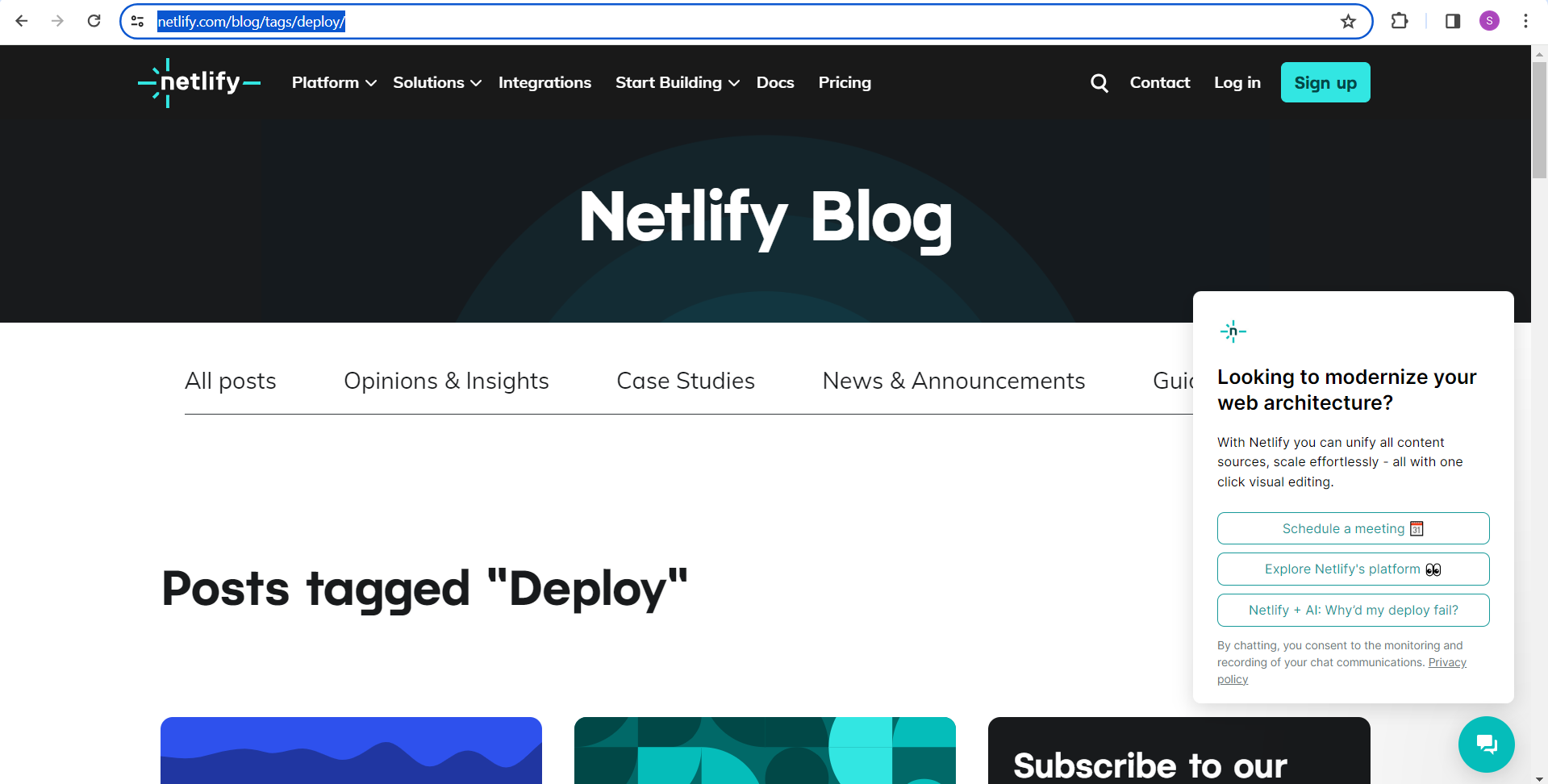
Pour chaque mise à jour du contenu, pour envoyer les modifications locales apportées au dépôt du serveur

git commit -a -m "Commit new version"

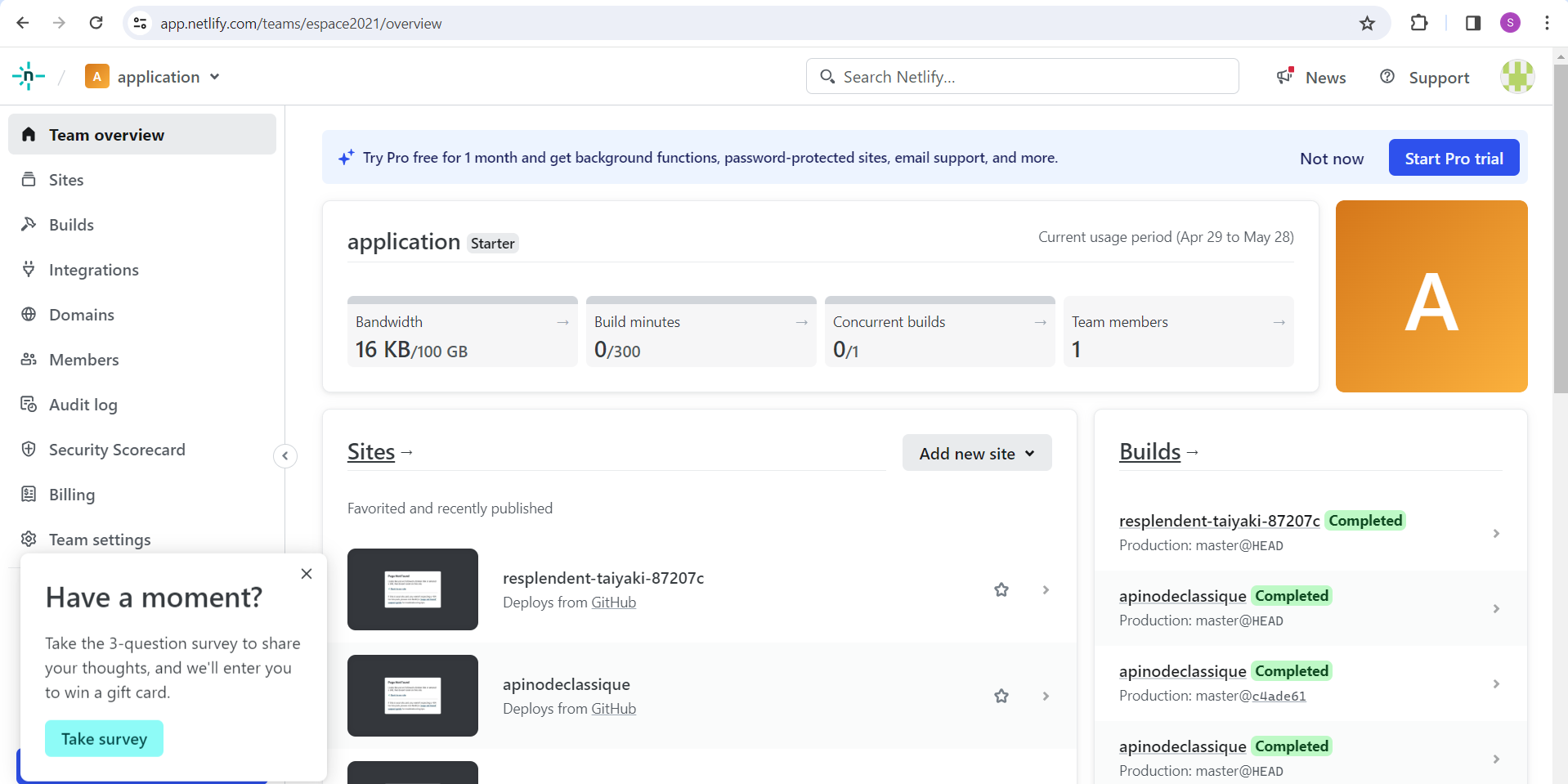
git push --set-upstream origin master

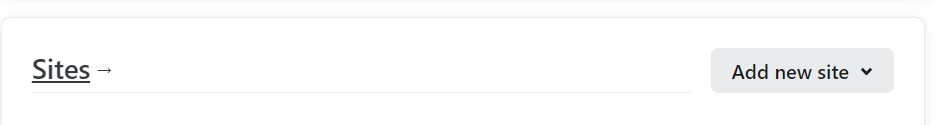
Netlify

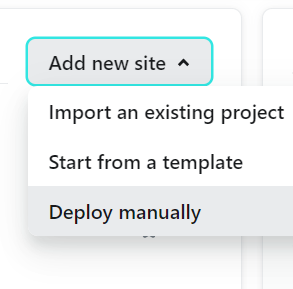
<https://www.netlify.com/blog/tags/deploy/>



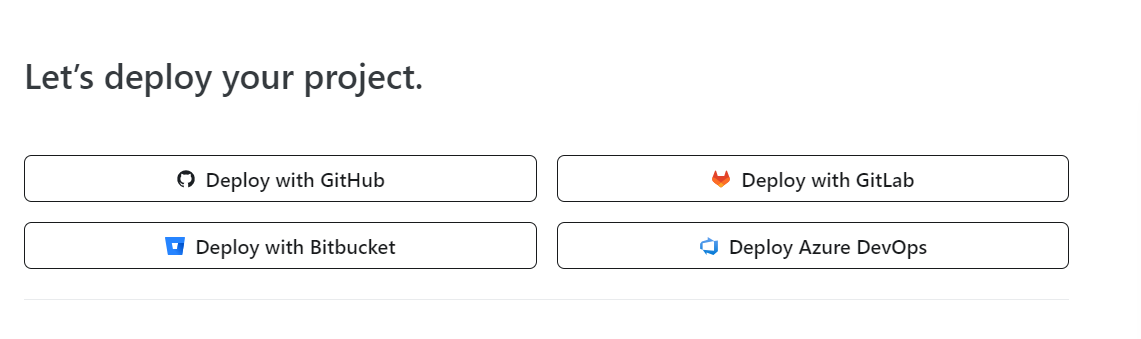


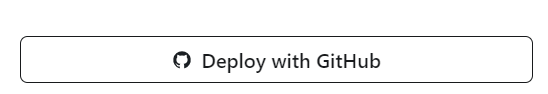


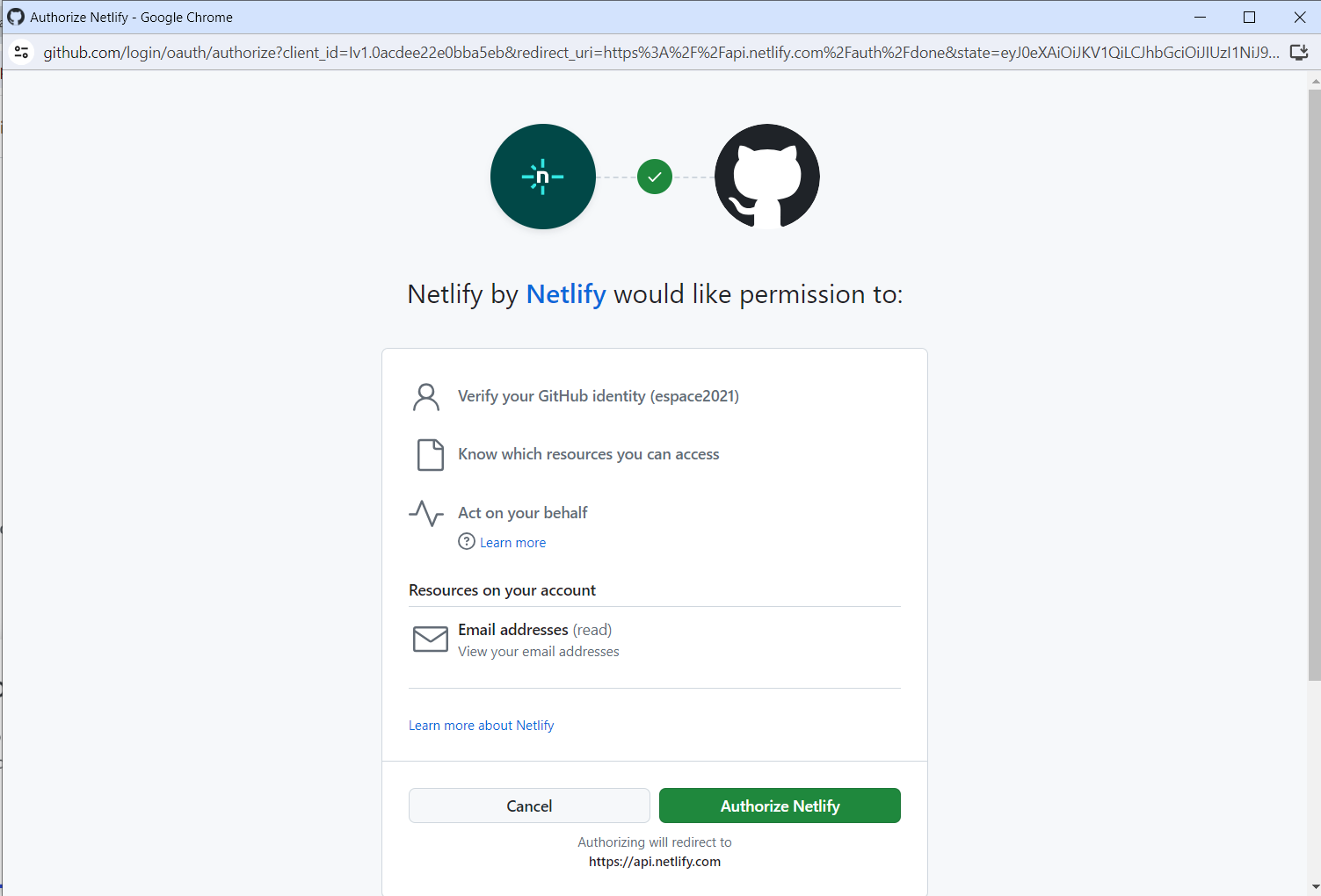


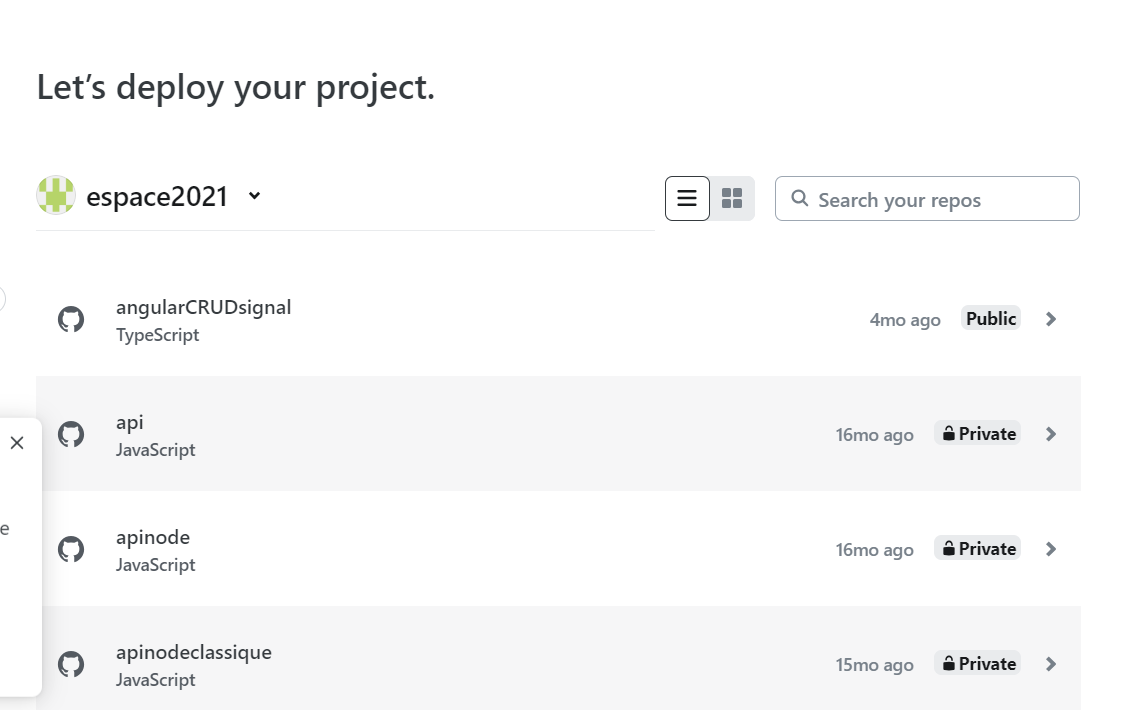


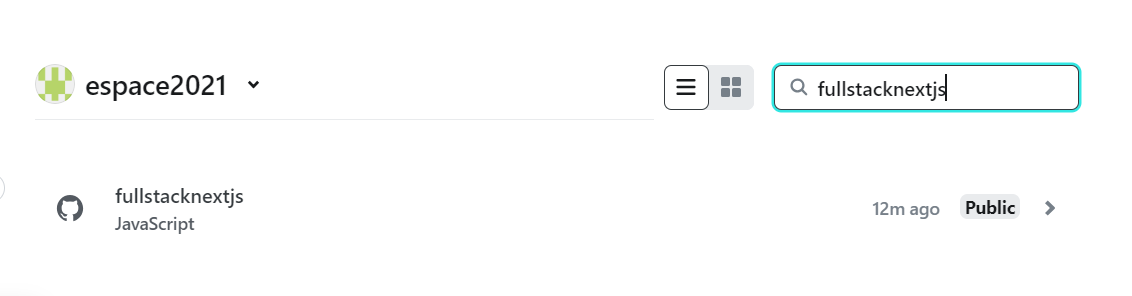


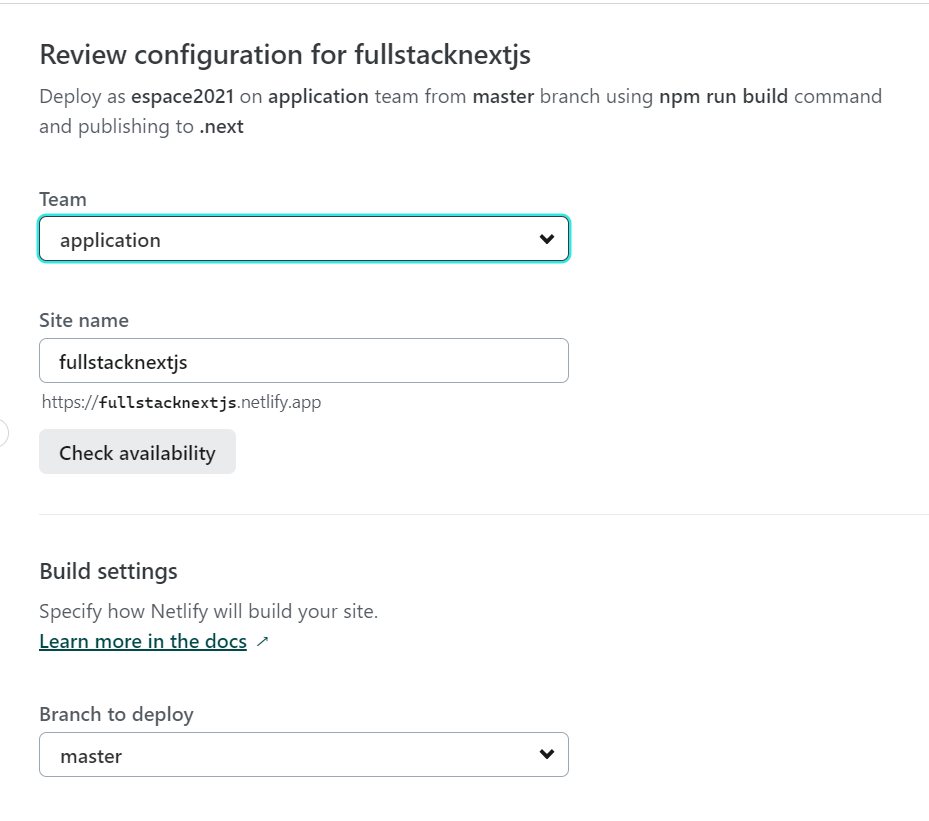


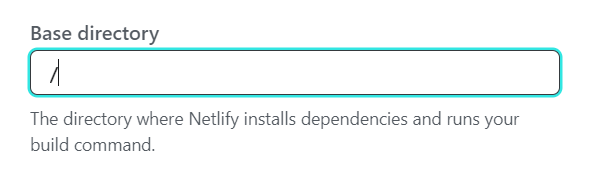


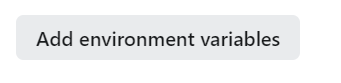


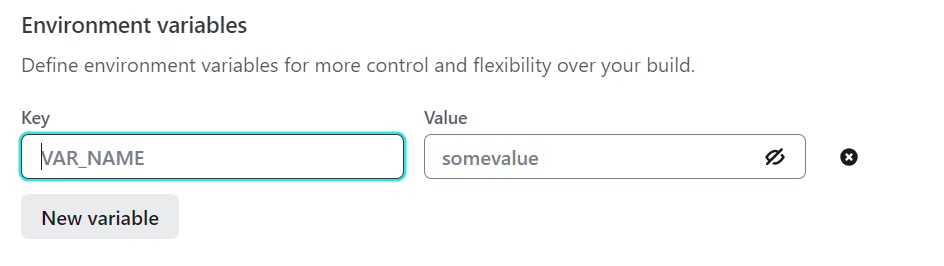


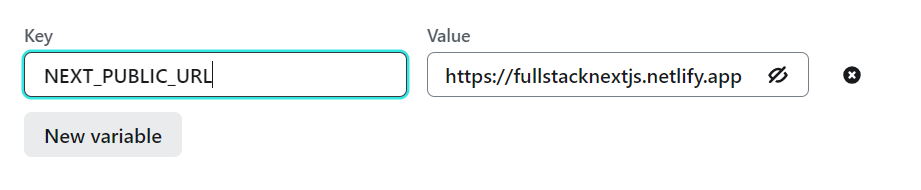


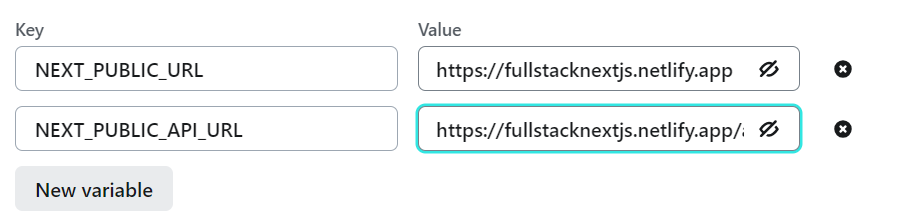










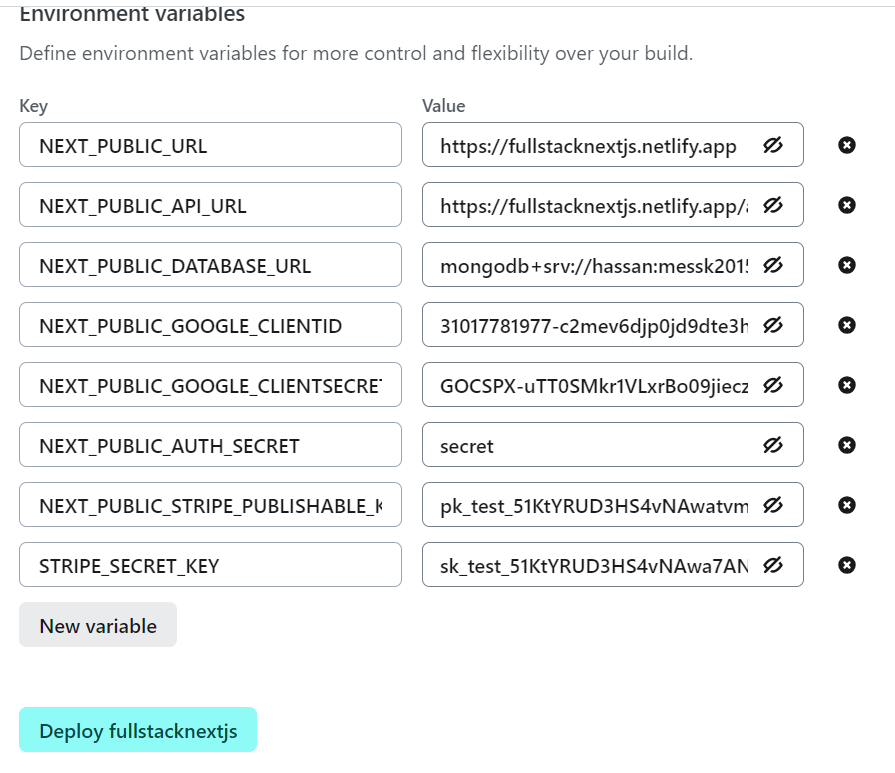


Vous y mettez comme valeur le nom du site à créer. Par exemple :

<https://fullstacknextjs.netlify.app>

et

<https://fullstacknextjs.netlify.app/api>



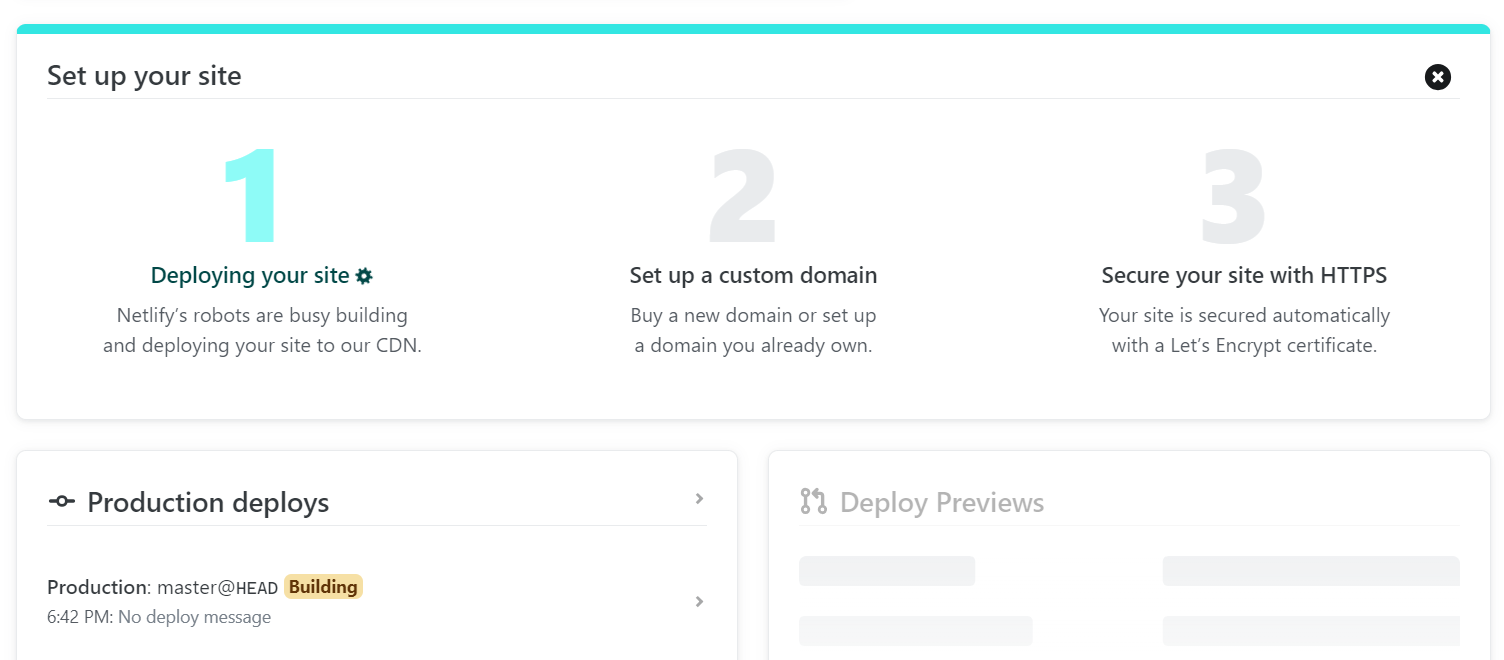
Enfin n’oubliez pas d’ajouter la variable NEXTAUTH\_URL ayant la valeur le nom du site à créer

Exemple :

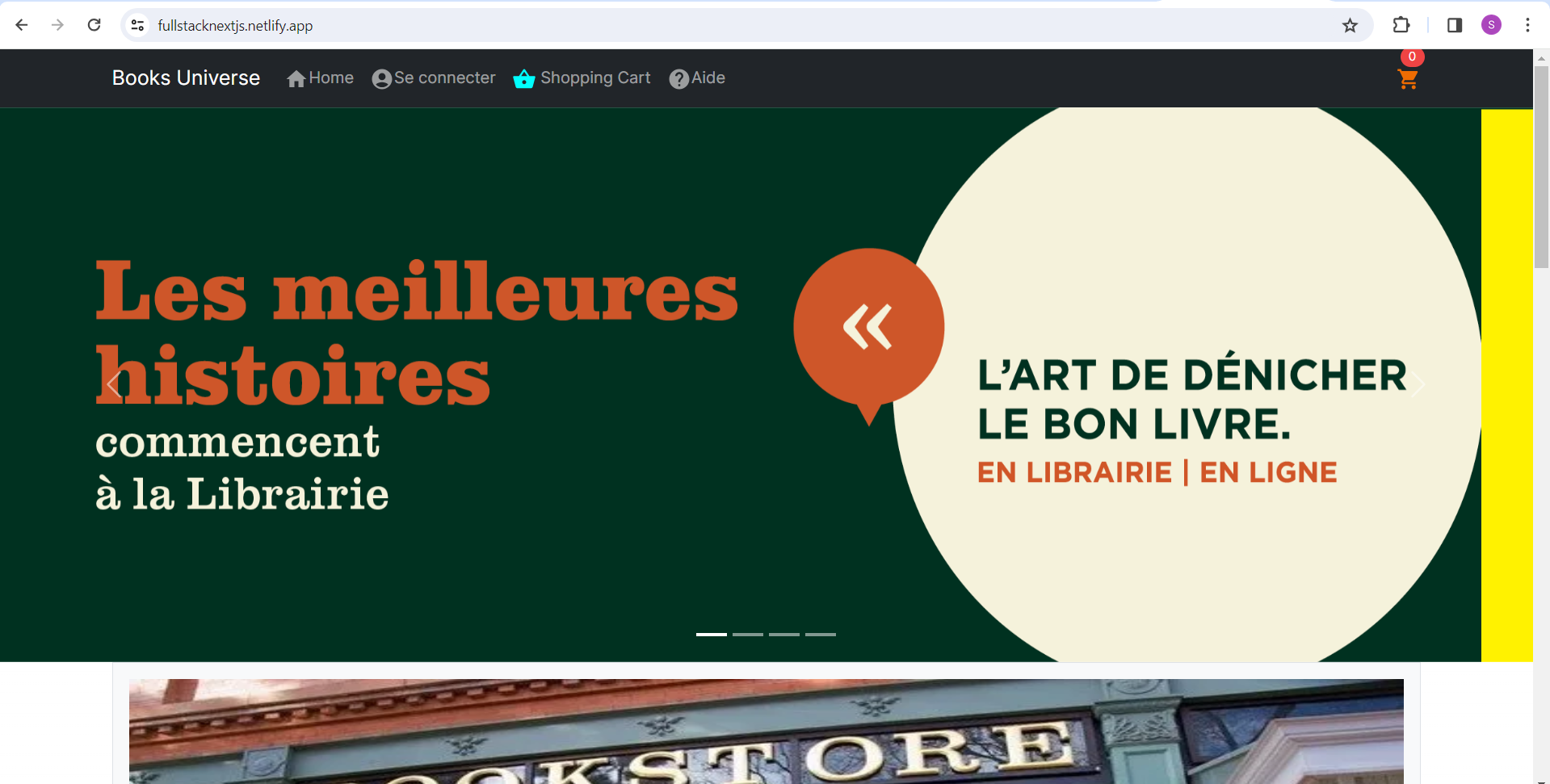
NEXTAUTH\_URL

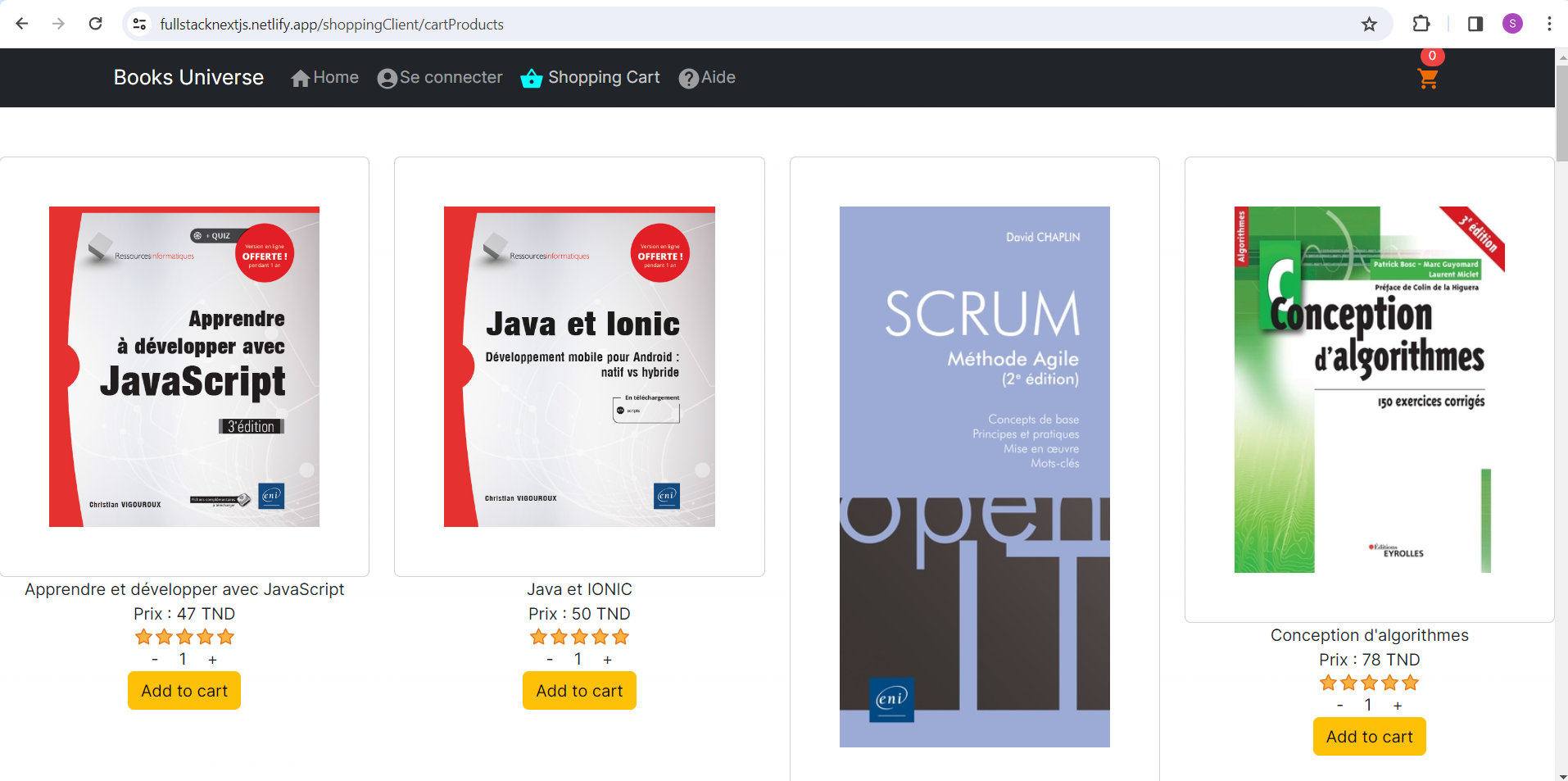
https://fullstacknextjs.netlify.app





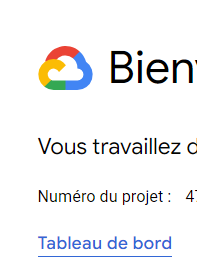
[https://fullstacknextjs.netlify.app](https://fullstacknextjs.netlify.app/)

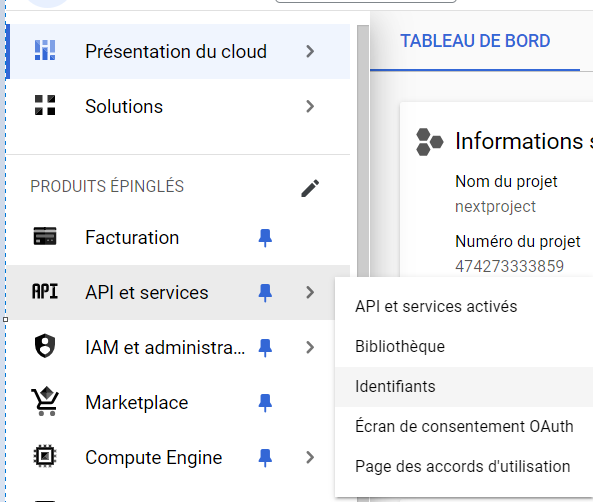


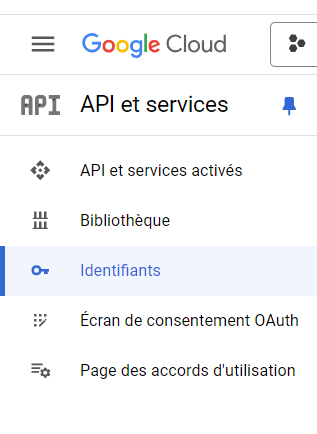


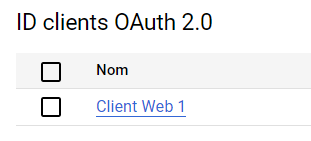
# Provider Google

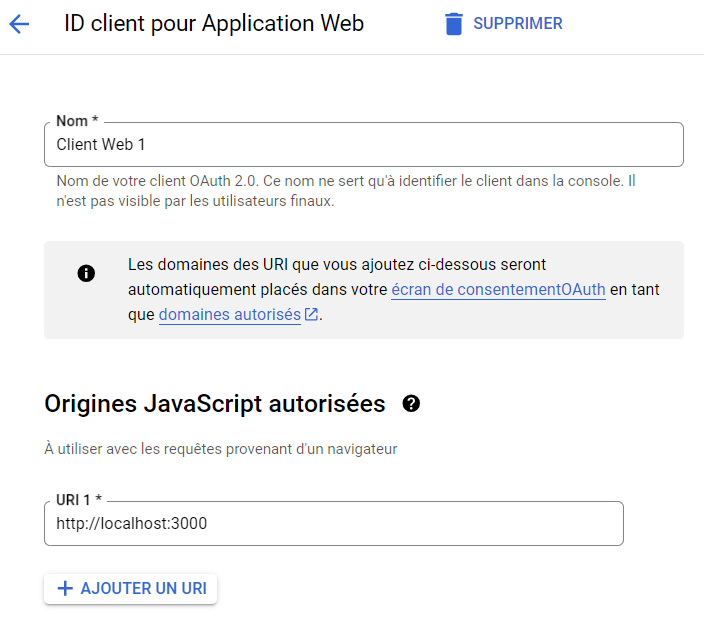
<https://console.cloud.google.com/>



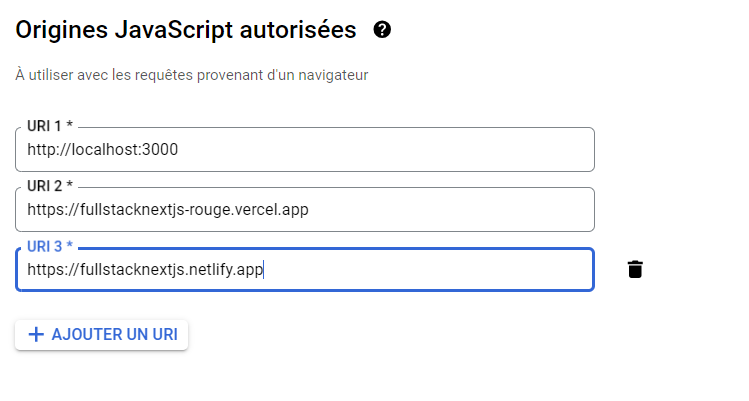


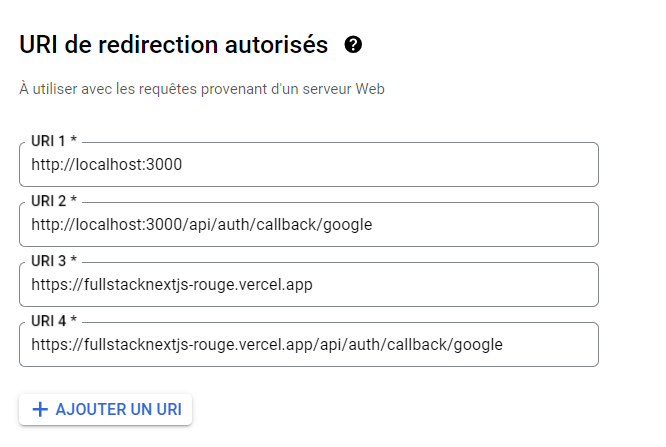




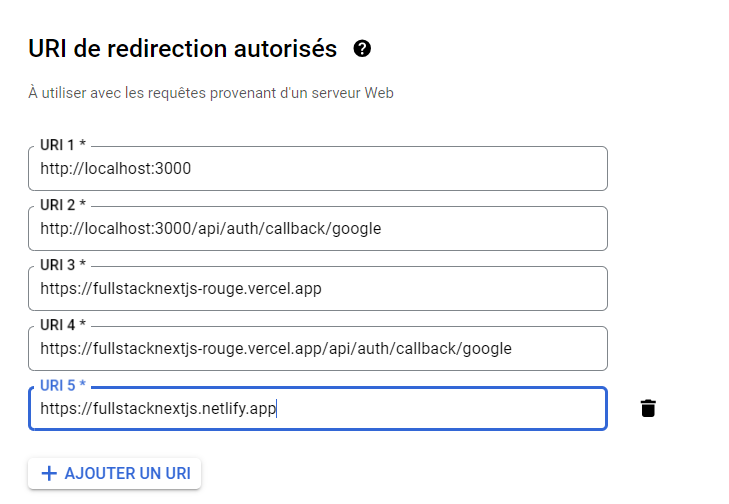








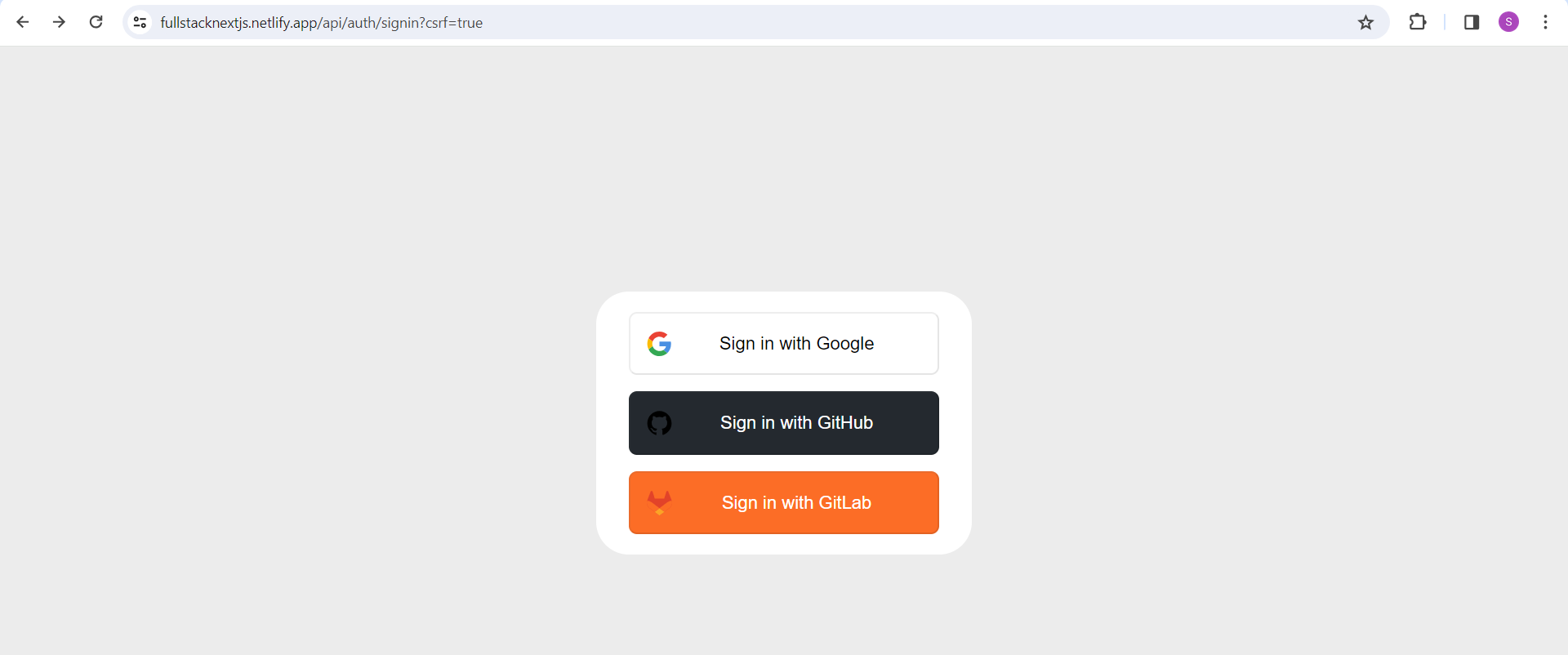


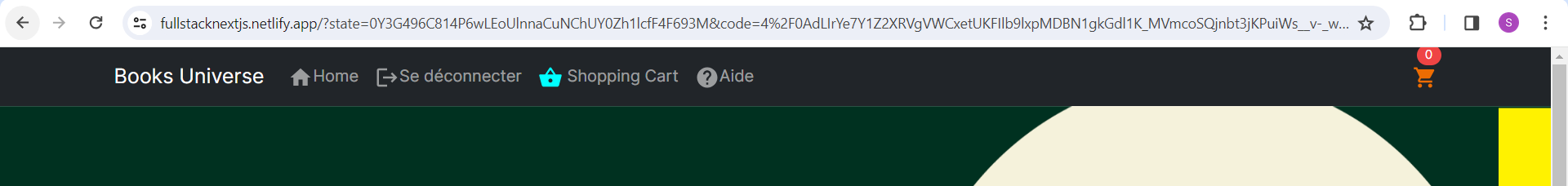














# Provider GitLab

<https://gitlab.com/oauth/applications/>

