Héberger son application chez Heroku

Le déploiement sur Heroku se fait en 2 étapes :

1. Déployer la base chez un hébergeur tel que mLab
2. Déployer l’application proprement dite sur Heroku

# *Héberger la BDD chez mLab :*

# *https://mlab.com/login/*

## Créer un compte « mLab »

- User / Pwd

## Définir un plan d'hébergement :

- SandBox

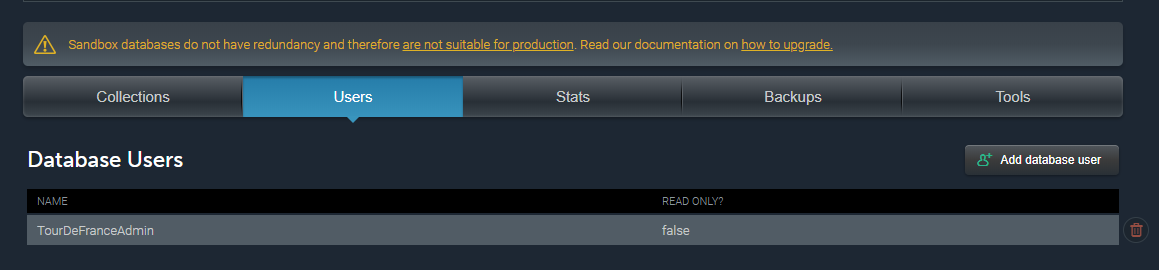
- Europe

- Amazon (AWS)

DB name : "tourdefrance"

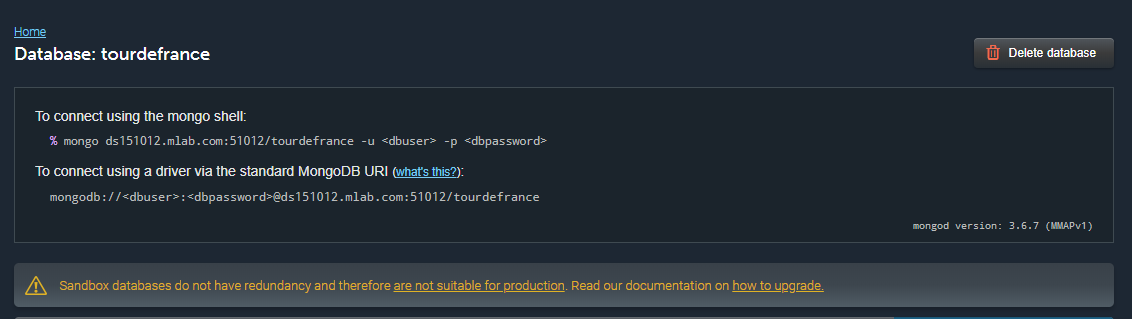
MongoDB : 3.6.7 (MMAPv1)

## Créer un user de la BDD :



- TourDeFranceAdmin / TDFAdmin\*001 (Admin de ma BDD)

## Noter les éléments de connexion fournis par mLab (dans le "Home" de la BDD créée)



**- a) Pour se connecter à un driver via l’URI standard MongoDB**

mongodb://TourDeFranceAdmin:TDFAdmin\*001@ds151012.mlab.com:51012/tourdefrance

**- b) Pour se connecter le Shell Mongol (Pour utilisation dans le mLab):**

mongo ds151012.mlab.com:51012/tourdefrance -u <dbuser> -p <dbpassword>

**- c) Pour anonymiser ses accès dans les chaines de connections (ce que je n’ai pas fait)**

SET MONGOLAB\_URI="mongodb://TourDeFranceAdmin:TDFAdmin\*001@ds151012.mlab.com:51012/tourdefrance

## Modifier la chaîne de connexion à la BDD dans l’application.

Utiliser la chaîne de connexion **(a)** fournie par mLab à l'instruction "connect" dans l'application

* ***Exemple*** *:*

Chaine de connexion **AVANT** hébergement sur mLAB dans l’application

…

mongoDB.MongoClient.connect("mongodb://localhost:27017/TourDeFrance", { useNewUrlParser: true }, (err,db) => {

…

Chaîne de connexion **APRES** hébergement sur mLAB dans l’application

…

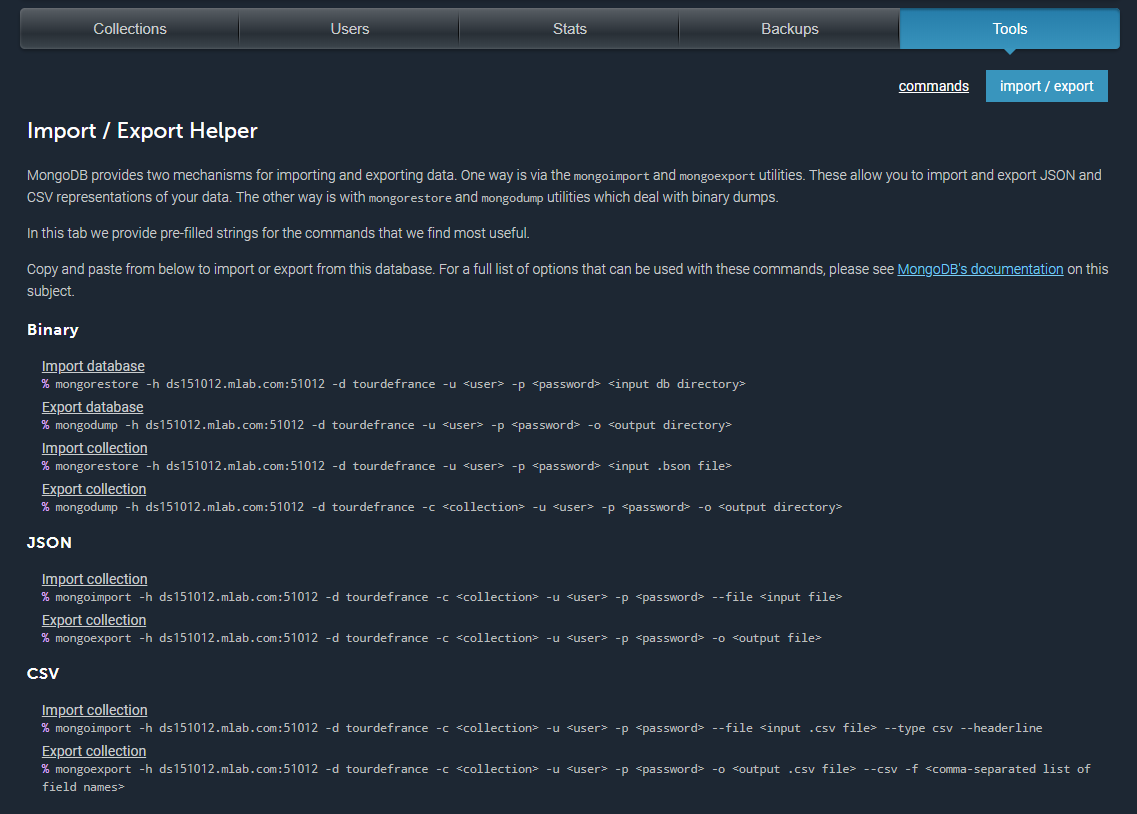
mongoDB.MongoClient.connect("mongodb://TourDeFranceAdmin:TDFAdmin\*001@ds151012.mlab.com:51012/tourdefrance", { useNewUrlParser: true }, (err,db) => {

…

|  |
| --- |
| **ATTENTION !!!**  **mLab n’accepte pas les majuscules dans les noms de BDD, aussi, corriger toutes les références aux noms avec majuscules en minuscule dans l’application, si nécessaire.** |

## Exporter / Importer

* Maintenant, il faut transférer les données de la base locale vers la base hébergée, qui existe, mais vierge



## a) Exporter la BDD locale (à exécuter dans un terminal type « cmd, « git bash », ou powerShell) :

***- mongoexport -d TourDeFrance -c joueurs -o ExportTourDeFrance.json***

## b) Importer la BDD dans mLab (à exécuter dans un terminal type « cmd, « git bash », ou powerShell) :

***mongoimport -h ds151012.mlab.com:51012 -d tourdefrance -c joueurs -u TourDeFranceAdmin -p TDFAdmin\*001 --file ExportTourDeFrance.json***

A ce niveau, le déploiement sur mLab est terminé, la base de données locale peut être éteinte (voire supprimée, après avoir testé le bon fonctionnement de l’application)

# *Déploiement de l’application sur Heroku*

# *https://id.heroku.com/login*

## Créer un compte « Heroku »

- User / Pwd

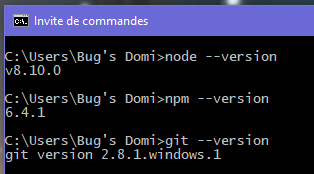
## Pré-requis  Dans un terminal (type « cmd, ou git bash) :

Les logiciels suivants doivent être installés sur le PC :

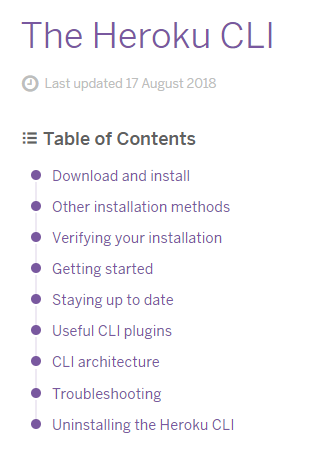
* Node.js
* Npm
* Git

Pour vérifier leur présence :

* node --version
* npm --version
* git –version

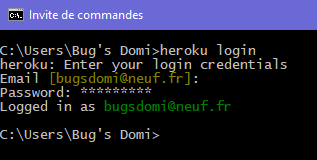


## Installer « Heroku CLI »

<https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

## Se logger à Heroku (Dans un terminal type « cmd, ou git bash)

* « **heroku login** »



## Aller dans le répertoire de l’application :

« **cd** D:\Cours Web\Cours\IFOCOP\Cours\DH\Ateliers\02 - Atelier Back\TourDeFrance>»

## Créer un fichier « Procfile » :

* dans le répertoire principal de l’application
* RESPECTER la casse du nom de fichier
* Pas d’extension de nom de fichier
* Ecrire « **web: node TourDeFrance.js** » dans ce fichier

## Modifier le Port de connexion du serveur dans l’application de la façon suivante :

Rajouter la variable « **process.env.PORT** » nécessaire à Heroku, car c’est lui qui gère le port du serveur à travers cette variable.

Déclaration du serveur **AVANT** déploiement sur Heroku

// -------------------------------------------------------------------------

// Création du serveur et lancement du listener

// -------------------------------------------------------------------------

const server = app.listen(3000, function() {

Déclaration du serveur **APRES** déploiement sur Heroku

// -------------------------------------------------------------------------

// Création du serveur et lancement du listener

// -------------------------------------------------------------------------

const server = app.listen(process.env.PORT || 3000, function() {

Noter que j’ai gardé le port d’origine derrière un opérateur « OU » (||), ce qui permet de continuer de lancer l’appli, hors Heroku

## Créer un repository Git

* **git init**

« D:\Cours Web\Cours\IFOCOP\Cours\DH\Ateliers\02 - Atelier Back\TourDeFrance> **git init** » 🡪

***Reinitialized existing Git repository in D:/Cours Web/Cours/IFOCOP/Cours/DH/Ateliers/02 - Atelier Back/TourDeFrance/.git/***

## Déployer l’application vers cet entrepôt (add, commit, push)

* **git add**

PS D:\Cours Web\Cours\IFOCOP\Cours\DH\Ateliers\02 - Atelier Back\TourDeFrance> **git add**

* **git commit -m "Deploy on Heroku - Ajout process.env.PORT"**

**(**La partie entre guillemet est personnelle et devra contenir ce que le déploiement corrige)

PS D:\Cours Web\Cours\IFOCOP\Cours\DH\Ateliers\02 - Atelier Back\TourDeFrance> **git commit -m "Deploy on Heroku - Ajout process.env.PORT" 🡪**

***On branch master***

***Your branch is up-to-date with 'origin/master'.***

***nothing to commit, working directory clean***

* **git push heroku master**

PS D:\Cours Web\Cours\IFOCOP\Cours\DH\Ateliers\02 - Atelier Back\TourDeFrance> **git push heroku master *🡪***

***Counting objects: 3, done.***

***Delta compression using up to 8 threads.***

***Compressing objects: 100% (3/3), done.***

***Writing objects: 100% (3/3), 340 bytes | 0 bytes/s, done.***

***Total 3 (delta 2), reused 0 (delta 0)***

***remote: Compressing source files... done.***

***remote: Building source:***

***remote:***

***remote: -----> Node.js app detected***

***remote:***

***remote: -----> Creating runtime environment***

***remote:***

***remote: NPM\_CONFIG\_LOGLEVEL=error***

***remote: NODE\_VERBOSE=false***

***remote: NODE\_ENV=production***

***remote: NODE\_MODULES\_CACHE=true***

***remote:***

***remote: -----> Installing binaries***

***remote: engines.node (package.json): unspecified***

***remote: engines.npm (package.json): unspecified (use default)***

***remote:***

***remote: Resolving node version 8.x...***

***remote: Downloading and installing node 8.11.4...***

***remote: Using default npm version: 5.6.0***

***remote:***

***remote: -----> Restoring cache***

***remote: - node\_modules***

***remote:***

***remote: -----> Building dependencies***

***remote: Installing node modules (package.json + package-lock)***

***remote: up to date in 1.08s***

***remote:***

***remote: -----> Caching build***

***remote: - node\_modules***

***remote:***

***remote: -----> Pruning devDependencies***

***remote: Skipping because npm 5.6.0 sometimes fails when running 'npm prune' due to a known issue***

***remote: https://github.com/npm/npm/issues/19356***

***remote:***

***remote: You can silence this warning by updating to at least npm 5.7.1 in your package.json***

***remote: https://devcenter.heroku.com/articles/nodejs-support#specifying-an-npm-version***

***remote:***

***remote: -----> Build succeeded!***

***remote: -----> Discovering process types***

***remote: Procfile declares types -> web***

***remote:***

***remote: -----> Compressing...***

***remote: Done: 30.1M***

***remote: -----> Launching...***

***remote: Released v4***

***remote: https://tourdefranceonheroku.herokuapp.com/ deployed to Heroku***

***remote:***

***remote: Verifying deploy... done.***

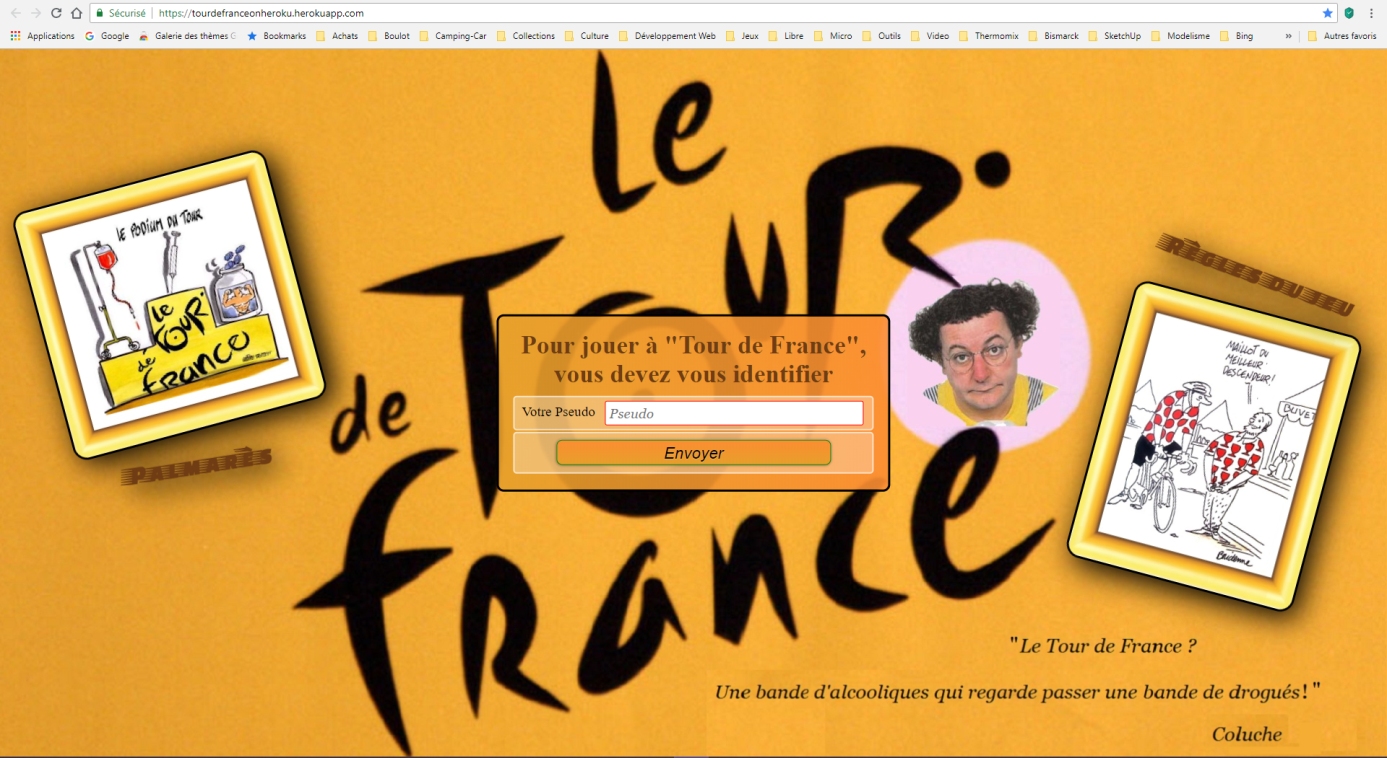
***To https://git.heroku.com/tourdefranceonheroku.git***

***8e66307..45f3455 master -> master***

***PS D:\Cours Web\Cours\IFOCOP\Cours\DH\Ateliers\02 - Atelier Back\TourDeFrance>***

## Lancer l’application hébergée sur Herooku

* **heroku open ou** taper l’adresse en jaune ci-dessus dans un navigateur



## Conserver précieusement les adresses suivantes

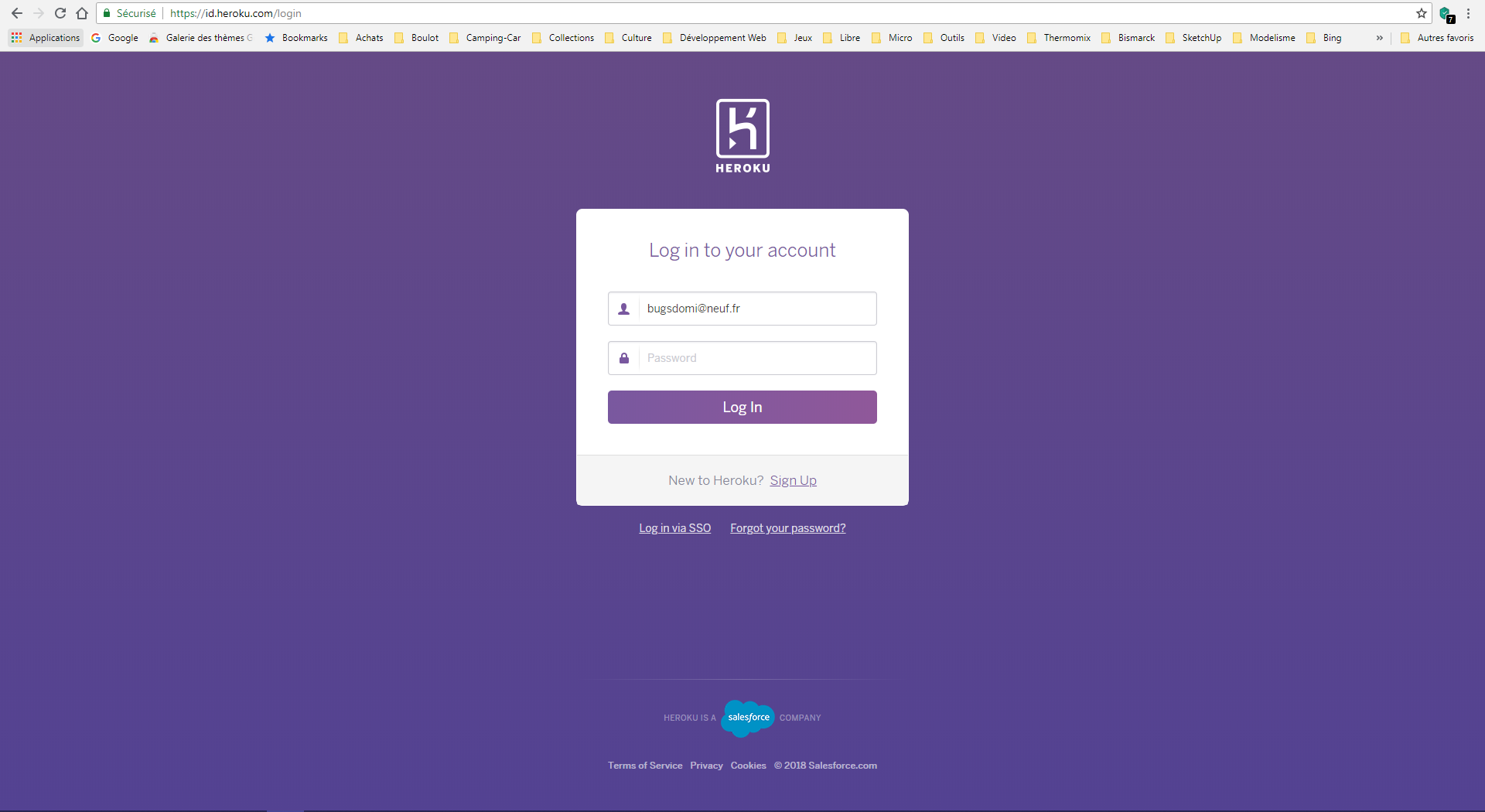
[**https://tourdefranceonheroku.herokuapp.com/**](https://tourdefranceonheroku.herokuapp.com/) **(URL de l’application hébergée sur heroku)**

[**https://git.heroku.com/tourdefranceonheroku.git**](https://git.heroku.com/tourdefranceonheroku.git) **(adresse dépôt Git sur Heroku)**

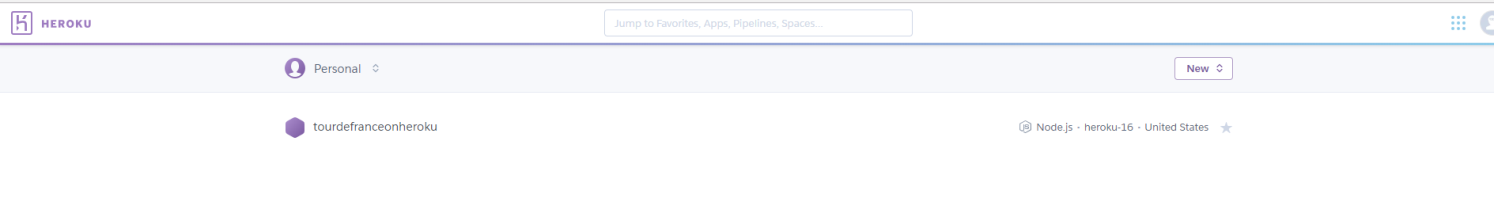
# Paramétrer un auto-déploiement du Dépôt Git de l’application vers Heroku

1. Se connecter à Heroku avec son compte

<https://id.heroku.com/login>

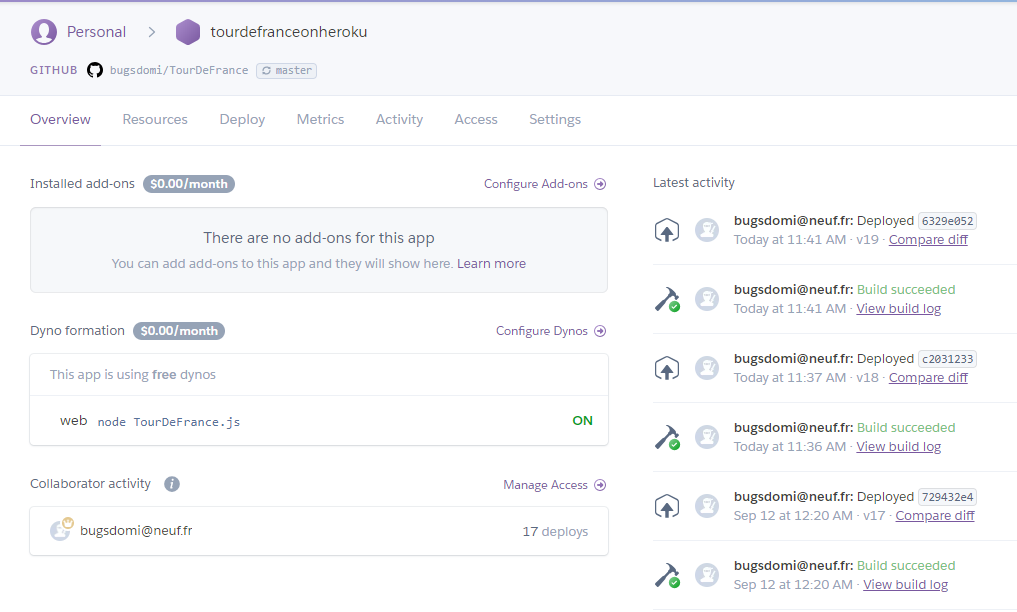


Cliquer sur l’application à auto-déployer

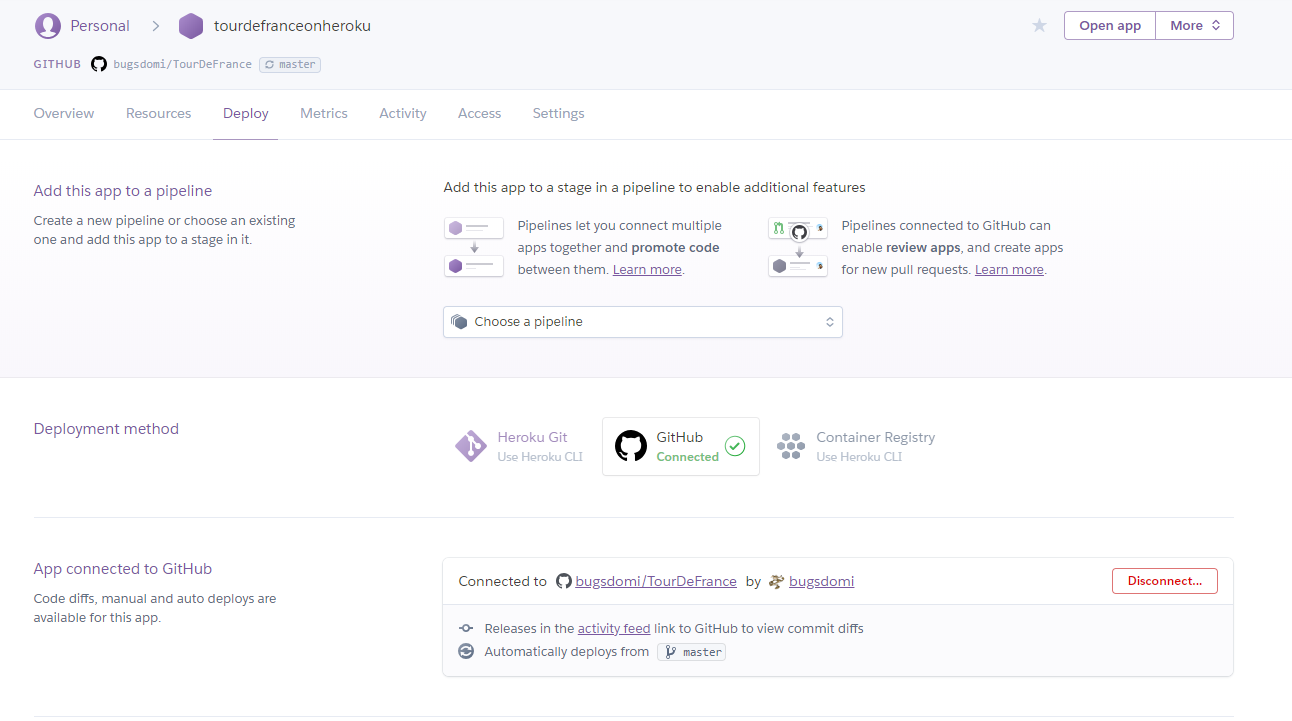
****

Ceci amène au DashBoard de l’application

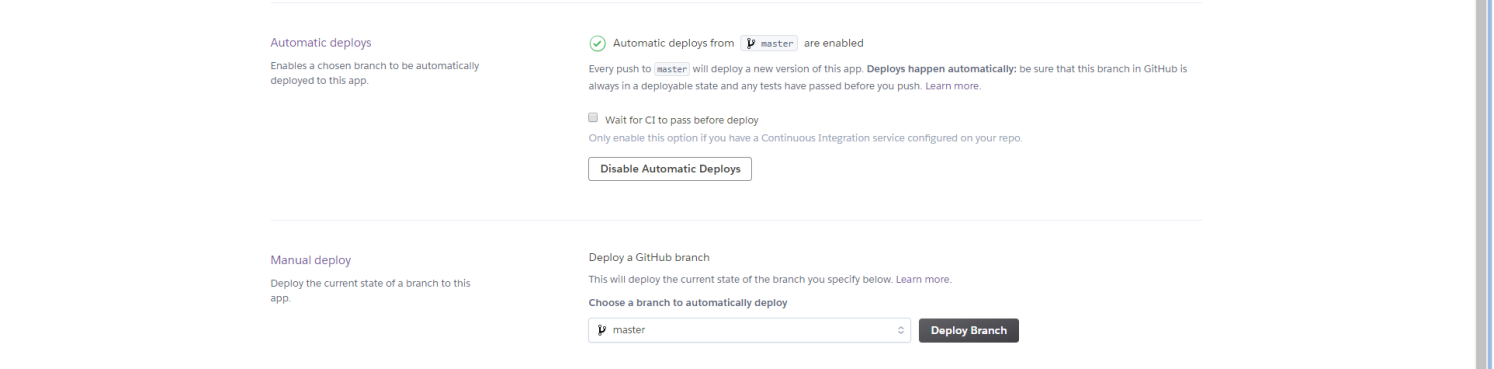
Selectionner l’onglet « Deploy »

****

Connecter le GitHub avec Heroku ici (si ce n’est pas déjà fait)



Activer le déploiement automatique et choisir la branche à deployer (normalement « master »)

****

Voila, c’est tout, et ça marche nickel !

Si nécessaire, il est possible de forcer le déploiement manuellement :

