1. Ouvrir un compte sur [www.heroku.com](http://www.heroku.com)
2. Télécharger et installer Git (<http://git-scm.com/download/win>)
3. Configurer Git
   1. git config --global user.email "[you@example.com](mailto:you@example.com)"
   2. git config --global user.name "Your Name"
4. Télécharger et installer Hero CLI (<https://cli-assets.heroku.com/heroku-x64.exe>)
5. Se logger à Heroku :
   1. Ouvrir la ligne de commande windows (cmd.exe)
   2. heroku login
   3. Appuyer sur une touche pour ouvrir une page web
   4. Sign in
6. Vérifier les versions :
   1. node --version
   2. npm --version
   3. git --version
7. Cloner localement une application test :
   1. git clone https://github.com/heroku/node-js-getting-started.git
   2. cd node-js-getting-started
8. Créer une application heroku

heroku create

Heroku attribut un nom aléatoire à votre application

1. Déploiement :

git push heroku master

Vérifier qu’au moins une instance de l’application s’exécute :

heroku ps:scale web=1

Tester l’application:

heroku open

1. Affichage des logs

heroku logs --tail

CTRL+C pour interrompre l’affichage

1. Définir le fichier Procfile

Echo web: node index.js > Procfile

1. Vérifier combien de dyno s’exécutent

Votre application s’exécute sur un seul dyno (dyno est un léger container qui exécute la commande spécifié dans Procfile)

heroku ps

1. Installer les dépendances localement

npm install

1. Exécuter l’application localement

heroku local web

Ouvrir la page  [http://localhost:5000](http://localhost:5000/) pour vérifier

CTRL+C pour arrêter l’exécution

1. Prendre en compte les changements

Exemple : installation d’un nouveau module

npm install cool-ascii-faces

modifier index.js en ajoutant les lignes en surbrillance

const cool = require('cool-ascii-faces')

const express = require('express')

const path = require('path')

const PORT = process.env.PORT || 5000

express()

.use(express.static(path.join(\_\_dirname, 'public')))

.set('views', path.join(\_\_dirname, 'views'))

.set('view engine', 'ejs')

.get('/', (req, res) => res.render('pages/index'))

.get('/cool', (req, res) => res.send(cool()))

.listen(PORT, () => console.log(`Listening on ${ PORT }`))

Tester localement

npm install

heroku local

Aller à la page :  <http://localhost:5000/cool>

Vous devez avoir ( ⚆ \_ ⚆ ).

Déploiement des changements :

git add .

git commit -m "Add cool face API"

git push heroku master

Pour tester:

heroku open cool

Ajout d’une base de données postgress

heroku addons:create heroku-postgresql:hobby-dev

npm install pg

Ajouter les lignes suivantes à index.js

const { Pool } = require('pg');

const pool = new Pool({

connectionString: process.env.DATABASE\_URL,

ssl: true

});

Et les lignes suivantes après .get('/', ...) :

.get('/db', async (req, res) => {

try {

const client = await pool.connect()

const result = await client.query('SELECT \* FROM test\_table');

const results = { 'results': (result) ? result.rows : null};

res.render('pages/db', results );

client.release();

} catch (err) {

console.error(err);

res.send("Error " + err);

}

})

Ce code retourne les enregistrements de la table test\_table , si vous utiliser la route /db

Déployer les changements (voir étape 15)

Également, si vous souhaitez utiliser votre nom de domaine ou un sous domaine, vous n’avez qu’à vous connecter au panneau de contrôle de Heroku (https://dashboard.heroku.com/apps) pour configurer le tout!

Source : https://devcenter.heroku.com/articles/getting-started-with-nodejs