

La mise en ligne d'applications web avec Heroku

Table des matières

I. Découvrez les bases d'Heroku	3
II. Exercice : Mettre en ligne une page HTML	6
III. Déployer et mettre en ligne une application web	6
IV. Exercice : Déployer une application avec une base de données	10
Solutions des exercices	10

I. Découvrez les bases d'Heroku

Durée : 1 h

Environnement de travail : Windows / (Mac OSX) / GNU / Linux

Pré-requis :

- Git
- Posséder un compte GitHub
- Utilisation basique ligne de commandes
- Savoir développer une application web

Contexte

Vous avez créé un site web et vous souhaitez le mettre en ligne, que ce soit pour le mettre à disposition de votre client ou tout simplement pour présenter un projet réalisé dans le cadre de votre formation.

Dans tous les cas, pour postuler à une offre d'emploi en développement web il est indispensable de savoir mettre en ligne son travail !

Mais s'il y a quelques années cette étape demandait de bonnes compétences en administration réseaux et quelques bases des systèmes d'exploitation GNU/Linux, aujourd'hui de nombreux services « *cloud* » facilitent drastiquement ce travail.

Heroku est l'une d'entre elles, et probablement la plus populaire de par sa facilité d'utilisation et son premier plan d'utilisation gratuit.

Pour suivre ce cours, il vous faudra quelques bases de git : savoir au minimum cloner un dépôt, sauvegarder votre travail sous forme de commits et savoir publier (ou « *push* ») votre branche de développement.

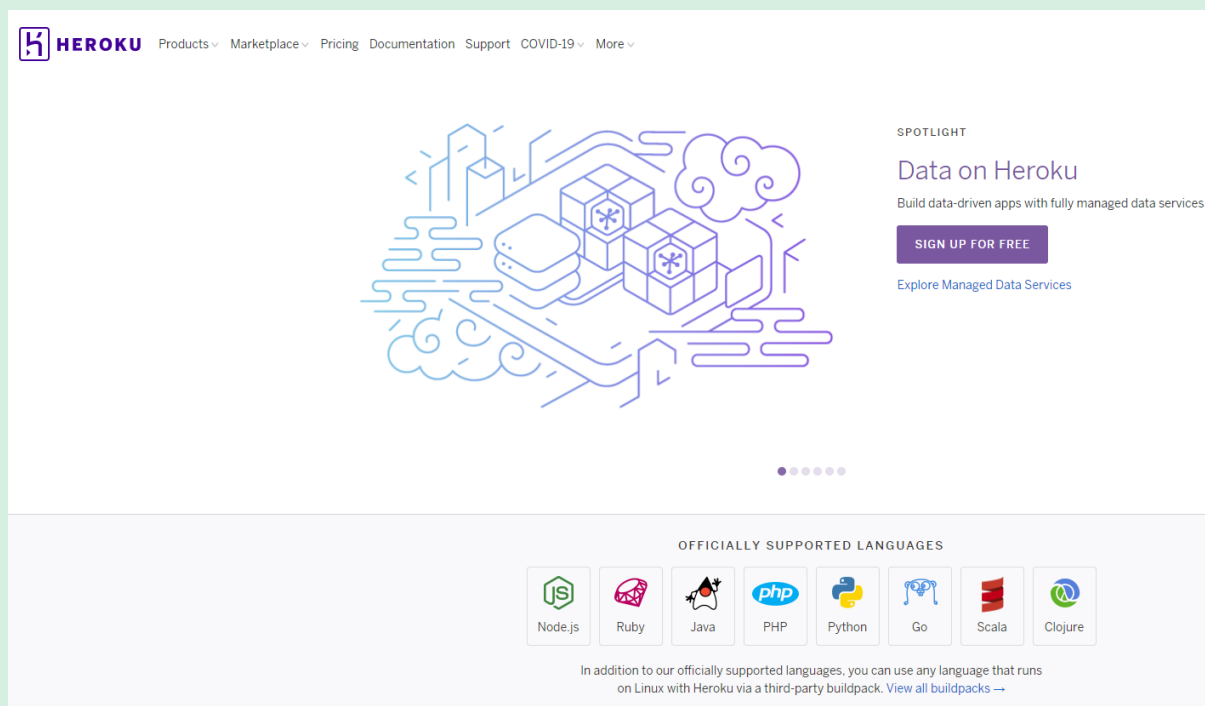
Dans moins d'une heure, vous aurez acquis les compétences suffisantes pour mettre votre application web en ligne.

Objectifs

- Installer Heroku / Activation Student Pack
- Découverte du client web et CLI
- Création du « *Hello world* »

Contexte

Heroku¹ est ce que l'on appelle « *une plateforme qui fournit un service* » ou PaaS pour « *platform as a service* » : c'est un site web qui fournit quelque chose de différent en termes de contenus qu'un site web classique.



Heroku est spécialisé dans le déploiement d'applications web, la gestion et l'exécution d'applications développées dans les langages suivants : Ruby, **Node.js**, Java, **Python**, Clojure, Scala, Go et **PHP**.

Heroku est aussi capable d'exécuter les images Docker, en théorie il peut donc exécuter n'importe quel langage de programmation.

La plateforme est « *intelligente* » : en fonction du code logiciel que vous allez vouloir déployer, Heroku va automatiquement exécuter certaines instructions comme l'installation des dépendances avec Composer pour PHP ou Npm pour Node.js.

Le déploiement des applications sur les serveurs mis à disposition d'Heroku sont effectués à l'aide du logiciel Git : ce n'est pas la *seule*² façon de faire, mais c'est la plus simple d'accès.

Dans ce chapitre, vous allez créer un compte sur Heroku et activer le plan **Hobby** à l'aide du crédit offert dans le « *Github Student Developer Pack*³ » disponible dans la section Studi Plus de la Médiathèque.

Ce plan vous donnera accès à de meilleures performances et plus de fonctionnalités.

Ensuite, vous allez découvrir l'interface de l'espace d'administration et installer le logiciel Heroku CLI.

Enfin, vous mettrez en ligne votre toute première application !

¹ <https://www.heroku.com/>

² <https://devcenter.heroku.com/articles/how-heroku-works#deploying-applications>

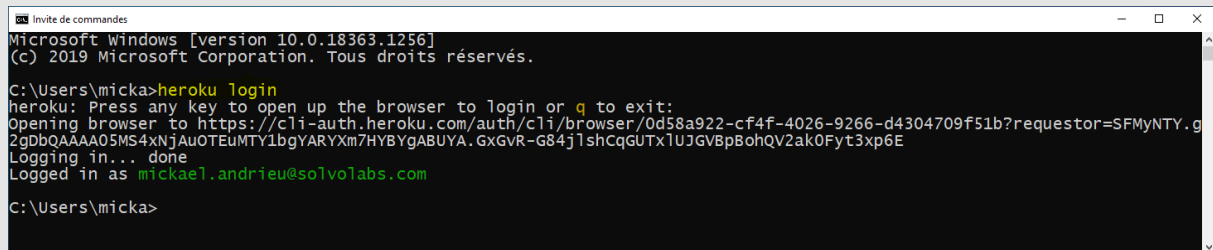
³ https://app.studi.fr/#/dashboard/medialib/acces-service/GITHUB_STUDENT_DEV

Méthode

Tout d'abord, il va falloir créer un compte sur la plateforme Heroku et installer Heroku CLI, pour cela suivez les instructions de la vidéo ci-dessous :

Heroku CLI est un logiciel qui s'utilise en ligne de commandes, il y a deux instructions à connaître pour pouvoir par la suite créer et déployer des applications : **heroku login** et **heroku create**.

La commande heroku login déclenche l'ouverture de votre navigateur et vous permet de vous authentifier à la plateforme depuis votre invite de commande. Il faudra toujours être authentifié avant d'exécuter d'autres instructions :



```
Microsoft Windows [version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\micka>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/0d58a922-cf4f-4026-9266-d4304709f51b?requestor=SFMyNTY.g
2gDbQAAAA05MS4xNjAuOTEuMTY1bGZARmYxM7HYBYgABUYA.GxGVR-G84j1shCqGUTx1UJGVBPBohQV2ak0Fyt3xp6E
Logging in... done
Logged in as mickael.andrieu@so1vo1abs.com

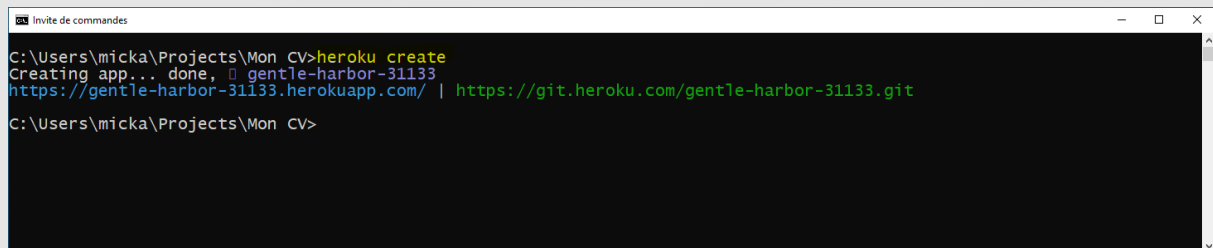
C:\Users\micka>
```

Ensuite, vous devez avoir un projet web à déployer ! Pour cela nous pouvons démarrer avec un simple fichier PHP de type Hello World appelé *index.php* :

```
<?php
echo("Hello
World");
```

Il faut que votre projet soit « versionné » à l'aide de git de sorte à pouvoir le publier par la suite sur le serveur que la plateforme Heroku vous mettra à disposition.

Ensuite, exécutez la commande heroku create : elle va créer une application sur Heroku (ou encore un « *dyno* »), ce qui consiste en un dépôt git et un sous domaine d'Heroku qui vous sera dédié :



```
C:\Users\micka\Projects\Mon CV>heroku create
Creating app... done, ⬢ gentle-harbor-31133
https://gentle-harbor-31133.herokuapp.com/ | https://git.heroku.com/gentle-harbor-31133.git

C:\Users\micka\Projects\Mon CV>
```

Il ne vous reste alors qu'à publier votre projet à l'aide de git comme vous le feriez pour déplacer vos fichiers sur GitHub !

```
git push heroku
master
```

Lors de cette action, Heroku détecte le type des fichiers envoyés et installe les logiciels nécessaires à leur exécution (PHP, Apache, Composer dans notre exemple).

Consultez les vidéos de démonstration ci-dessous pour voir l'exécution de ces commandes dans le cadre d'une mise en ligne du projet.

Syntaxe **À retenir**

L'intérêt de la plateforme Heroku tient dans sa simplicité d'utilisation. En se basant sur Git et après avoir installé le logiciel Heroku CLI, il suffit de retenir deux instructions pour être capable de déployer un projet simple sur internet :

- Heroku login : qui vous authentifie à la plateforme depuis votre invite de commandes,
- Heroku create : qui crée le dépôt git où déployer vos fichiers et l'URI où l'application sera accessible depuis un navigateur web.

Et ... c'est tout ! Nous verrons dans le prochain chapitre comment déployer des applications plus complexes.

Complément

- Heroku CLI (La documentation officielle)^{[EN]1}
- Renommer ses Apps avec Heroku CLI²
- S'authentifier avec Heroku CLI³

II. Exercice : Mettre en ligne une page HTML

Question

Et si vous mettiez en ligne votre CV ? Si vous n'en avez pas à disposition, vous pouvez vous servir d'un modèle disponible sur GitHub à cette adresse⁴. Pensez à en adapter le contenu et à adapter l'URL de l'application de sorte à pouvoir mettre ce lien sur vos CV au format papier ou PDF.

Indice :

Si ce n'est pas encore le cas, il faut gérer (ou encore « *versionner* ») son CV à l'aide de Git, puis créer une application Heroku, puis déployer sur le serveur d'Heroku à l'aide de Git.

Tout a été vu dans les vidéos précédentes, cet exercice vous permet d'appliquer ce qui a été vu.

Mais il y a peut-être plus simple⁵ à faire ...

III. Déployer et mettre en ligne une application web

Objectifs

- Mettre en place les addons nécessaires
- Le fichier ProcFile
- Les variables d'environnement

Contexte

Avec Heroku et en quelques instructions vous avez réussi à mettre en ligne un script, puis un CV en ligne ... mais les applications et sites web d'aujourd'hui sont plus complexes que cela !

Comment configurer une base de données, ou passer des clés d'API ou configurer le point d'entrée de l'application si le fichier principal n'est pas à la racine du projet ou encore, qu'il n'est pas reconnu par Heroku ?

¹ <https://devcenter.heroku.com/articles/heroku-cli>

² <https://devcenter.heroku.com/articles/renaming-apps>

³ <https://devcenter.heroku.com/articles/heroku-cli#:~:text=with%20these%20instructions.,Getting%20started,then%20log%20you%20in%20automatically>

⁴ <https://github.com/mickaelandrieu/little-cv-app>

⁵ <https://devcenter.heroku.com/articles/creating-apps#creating-a-named-app>

Dans ce chapitre, vous allez installer des add-ons. Ce sont des briques logicielles que l'on peut ajouter à son projet et qui fournissent tous les éléments nécessaires au fonctionnement de votre projet.

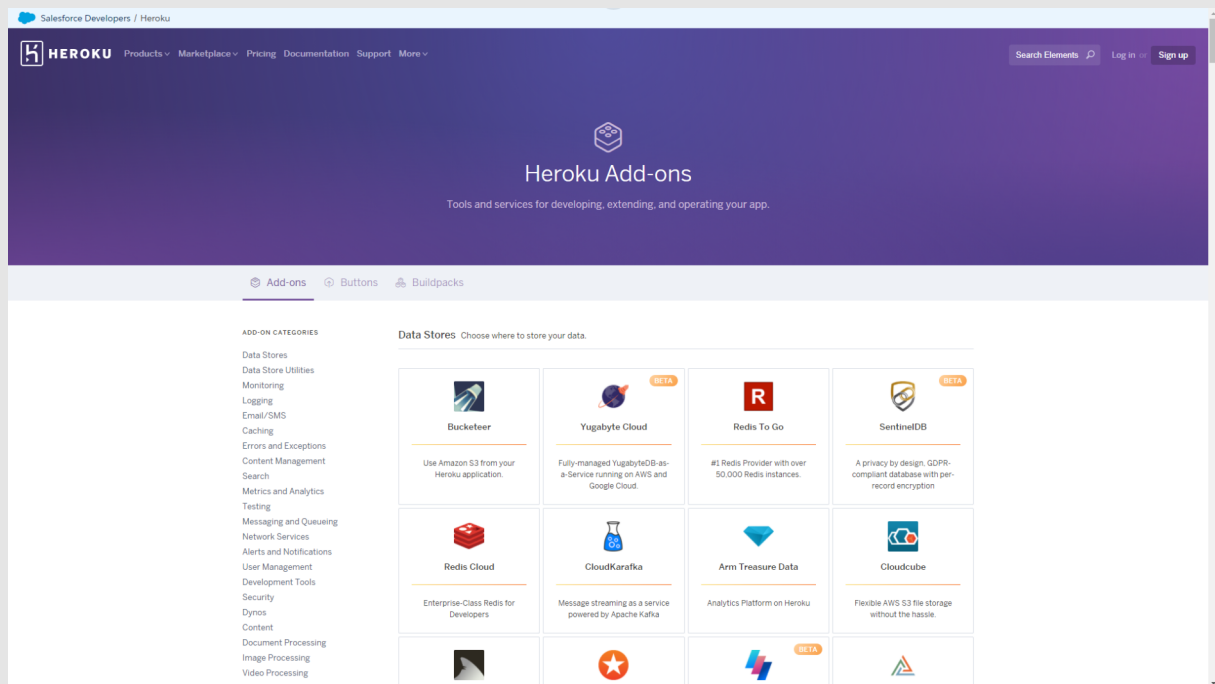
Vous allez mettre en place un fichier *ProcFile* pour adapter le comportement d'Heroku lors du déploiement de votre application : rien de compliqué, mais parfois nécessaire pour les projets les plus populaires.

Enfin, vous apprendrez à passer votre configuration de projet en tant que variables d'environnement de sorte à conserver un projet fonctionnel en local, et sur Heroku !

Ce chapitre se termine sur des démonstrations de mise en ligne de projets populaires en JavaScript et PHP.

Méthode Addons

Les add-ons ou encore « *Elements*¹ » vont vous permettre d'ajouter des logiciels supplémentaires comme par exemple une base de données à votre projet.



Pour ajouter un add-on à une application, nous utiliserons encore une fois Heroku CLI :

```
heroku addons:create <nom-addon> --app <nom-app>
```

Par exemple :

```
heroku addons:create heroku-redis --app mon-super-cv
```

¹ <https://elements.heroku.com/addons>

Consultez la vidéo de démonstration ci-dessous pour voir comment mettre en place MySQL dans une application PHP classique :

Méthode Le fichier ProcFile

La plateforme Heroku adopte un comportement en fonction des fichiers qu'elle reçoit. Par exemple, elle va exécuter le fichier PHP `index.php` si elle le trouve et considérer que c'est d'un environnement PHP dont vous avez besoin.

Mais que se passe-t-il si le fichier principal est plutôt dans un dossier spécifique comme « `web` » ou « `public` » ?

On doit alors ajouter un fichier appelé *ProcFile*¹ pour définir le comportement d'Heroku.

Ce fichier peut permettre de configurer d'autres processus mais dans le cadre de ce cours, nous nous limiterons à définir la commande à exécuter pour démarrer votre application web.

L'instruction à écrire dans ce fichier qui n'a pas d'extension est de la forme suivante :

```
web: <instruction de démarrage de
l'application>
```

Cette instruction dépendra évidemment de votre besoin, et il faudra chercher dans la documentation d'Heroku comment faire selon le type d'application.

Par exemple, si l'on souhaite définir le fait que le serveur web aille chercher le point d'entrée de l'application web dans le dossier `public` plutôt qu'à la racine du projet alors on définit l'instruction suivante :

```
web: vendor/bin/heroku-php-apache2
public/
```

Mais s'il n'y a plus de fichier `index.php`, comment Heroku va-t-il détecter que l'application à déployer est une application PHP ?

Voici la liste des fichiers recherchés par la plateforme Heroku pour détecter l'environnement logiciel à mettre en place :

Langage	Fichiers	Documentation
PHP	<code>index.php</code> ; <code>composer.json</code>	Heroku PHP Support
Node.js	<code>package.json</code>	Heroku Node.js Support
Python	<code>requirements.txt</code> ; <code>setup.py</code> ; Pipfile	Heroku Python Support ²
Java	<code>pom.xml</code>	Heroku Java Support ³

¹ <https://devcenter.heroku.com/articles/procfile>

² <https://devcenter.heroku.com/articles/python-support>

³ <https://devcenter.heroku.com/articles/java-support>

Pour voir un exemple de mise en place du fichier ProcFile, consultez la vidéo ci-dessous :

Méthode Les variables d'environnement

Les applications modernes se configurent à l'aide de variables d'environnement. Dans heroku CLI, vous pouvez accéder aux variables d'environnement disponibles pour une application à l'aide de l'instruction **heroku config**.

Pour créer et assigner la valeur d'une variable d'environnement, on utilisera l'instruction suivante :

```
heroku config:set <nom-variable>=<valeur>
```

Par exemple, pour assigner une variable d'environnement « APP_ENV » avec la valeur « PRODUCTION » :

```
heroku config:set  
APP_ENV=PRODUCTION
```

Une autre commande pour lister toutes les variables d'environnement (dont celles de la plateforme Heroku) est la suivante :

```
heroku run  
printenv
```

On y retrouvera une information importante pour tous les développeurs et développeuses Node.js : le port HTTP utilisé sur l'environnement Heroku ! Vous pourrez donc configurer dans vos applications le port de la façon suivante :

```
const port = process.env.PORT ||  
3000
```

Consultez la vidéo ci-dessous pour un exemple d'utilisation et de configuration des variables d'environnement dans Heroku :

Exemple Code

Pour un complément d'informations sur la mise en place d'une application Symfony, l'un des meilleurs tutoriels sur le sujet est celui publié sur le site de l'association française¹ du framework.

Syntaxe À retenir

Heroku est tout à fait adapté au déploiement d'applications plus complexes, qui ont besoin de plusieurs logiciels et d'une configuration qui évolue entre la mise en production et l'environnement local.

D'abord, les Addons proposés sur le site Elements², ils s'installent en une commande et se configurent à l'aide de variables d'environnement. Ils peuvent être gratuits, ou payants et sont disponibles pour la plupart avec de nombreux « plans » de tarification en fonction de vos besoins.

Vous utiliserez l'instruction **heroku addons:create** suivi du nom du module et son plan.

Ces variables d'environnement sont disponibles et définies à l'aide d'Heroku CLI ou dans l'espace d'administration : elles permettent de configurer les éléments importants de vos applications.

La base de données, le mode d'environnement, des clés d'API pour une authentification ... tout peut être passé à Heroku en toute sécurité.

¹ <https://afsy.fr/avent/2017/03-deployer-un-projet-symfony-flex-sur-heroku>

² <https://elements.heroku.com/>

Pour consulter les variables d'environnement disponibles, l'instruction la plus utile est **heroku run printenv** et vous définirez de nouvelles variables à l'aide d'**heroku config:set**.

Pour conclure, il faut parfois altérer le comportement par défaut d'Heroku pour les besoins spécifiques d'une application : il faudra alors proposer un fichier **Procfile** à la racine du projet dans lequel on définit l'instruction à exécuter au démarrage de l'application.

Il vous reste encore énormément de choses à découvrir et à apprendre sur Heroku, mais vous avez maintenant toutes les informations nécessaires pour mettre en ligne vos projets scolaires !

L'exercice final vous fera découvrir une autre fonctionnalité d'Heroku : le déploiement d'applications en 1 clic.

Complément

- Gestion des variables d'environnement^{[EN]1}
- Le fichier Procfile^{[EN]2}
- La gestion des addons^{[EN]3}

IV. Exercice : Déployer une application avec une base de données

Question

Après avoir lu la documentation sur les « *Buttons* », recherchez sur Heroku Elements et déployez en 1 clic une application WordPress.

Indice :

Il faut accéder à ce site, et rechercher « *WordPress* » ... prenez l'application la plus populaire, c'est souvent la meilleure !

Solutions des exercices

¹ <https://devcenter.heroku.com/articles/config-vars>

² <https://devcenter.heroku.com/articles/procfile>

³ <https://devcenter.heroku.com/articles/managing-add-ons>

Exercice p. Solution n°1

Si vous êtes partis du projet fourni, voici les commandes qui permettent la mise en ligne du CV :

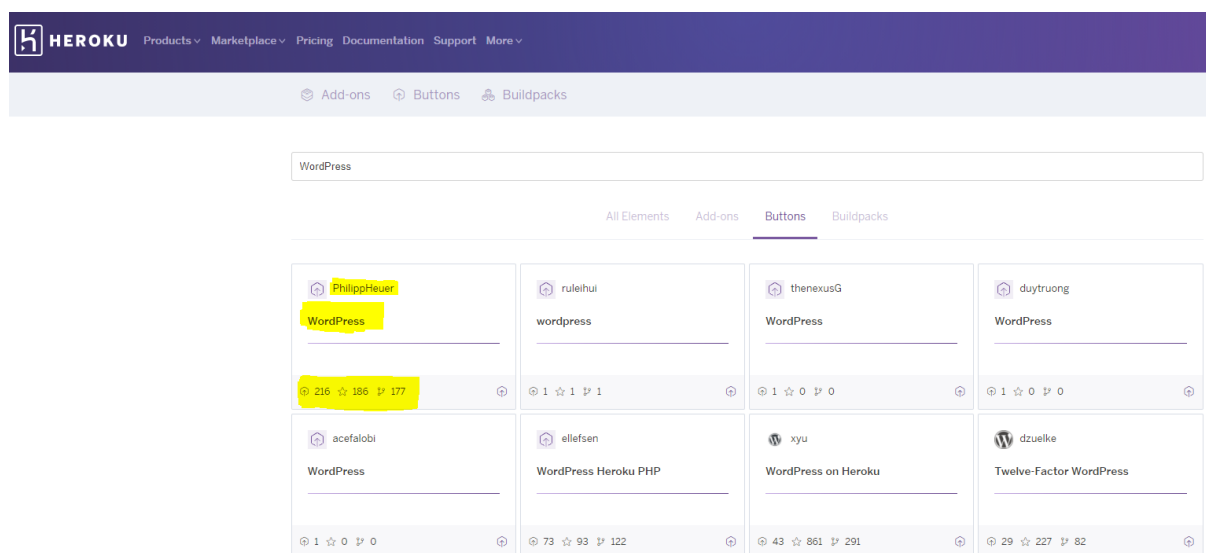
```
git clone mickaelandrieu / little-cv-app.git
heroku create mon-super-cv
git push heroku master
```

Attention ! Le nom « *mon-super-cv* » n'est plus disponible, vous devrez donc trouver le vôtre.

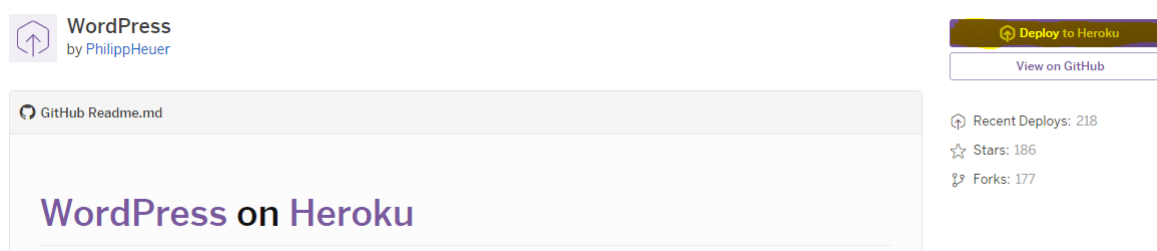
Le projet d'exemple est accessible à cette adresse herokuapp¹

Exercice p. Solution n°2

Après avoir effectué votre recherche tel que proposé dans l'indice, cliquez sur le bloc en surbrillance :



Puis sur « *Deploy on Heroku* » :



Enfin, donnez un nom à votre application et dans « *Region* » choisissez « *Europe* » puis tout en bas de la page, cliquez sur le bouton « *Deploy app* ».

¹ <https://mon-super-cv.herokuapp.com/>



Deploy your own

[WordPress](#)

WordPress is open source software you can use to create a beautiful website, blog, or app.

[PhilippHeuer/wordpress-heroku#master](#)

App name



test-wp-studi is available

Choose a region

Europe



Add to pipeline...

Add-ons

These add-ons will be provisioned when the app is deployed.



JawsDB Maria

Kitefin Shared

Free



SendGrid

Starter

Free



Heroku Scheduler

Standard

Free



Heroku Redis

Hobby Dev

Free



Papertrail

Choklad

Free

Félicitations ! Votre application WordPress (après quelques minutes sera disponible à l'URI disponible dans la partie Settings de l'application créée. Vous venez de déployer une application complète et fonctionnelle en un seul clic.