

ArchitectApp

Dossier d'exploitation

Version 1.0

Auteur

PANCARTE Marjorie
Analyste Développeuse

TABLE DES MATIERES

1 - Versions	3
2 - Introduction	4
2.1 - Objet du document	4
2.2 - Références	4
3 - Pré-requis	5
3.1 - Système	5
3.2 - Bases de données	5
3.3 - Web-services	Erreur ! Signet non défini.
4 - Procédure de déploiement	6
4.1 - Déploiement de l'application web	6
4.2 - Déploiement de la base de données	7
5 - Procédure de démarrage / arrêt	10
5.1 - Démarrage / arrêt sur le site heroku.com	10
5.2 - Démarrage / arrêt en ligne de commande	10
6 - Procédure de mise à jour	11
7 - Supervision/Monitoring	12
7.1 - Supervision de l'application web	12
8 - Procédure de sauvegarde et restauration	13
9 - Glossaire	14

1 - VERSIONS

Auteur	Date	Description	Version
PM	30/07/2019	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application Architect.

L'objectif du document est de fournir à l'équipe les informations essentielles pour une bonne utilisation de l'application, ainsi que les instructions à suivre pour le déploiement et la maintenance de celle-ci.

2.2 - Références

Pour de plus amples informations, se référer :

1. **P8 DCT – 1.1**: Dossier de conception technique de l'application
2. **P8. DCF – 1.1** : Dossier de conception fonctionnelle de l'application

3 - PRÉ-REQUIS

3.1 - Système

L'application web Architect est hébergé sur la plateforme « Heroku ».

L'application est liée au nom de domaine : « architectapp.fr »

3.2 - Bases de données

Le SGBD utilisé par l'application est MySQL dans sa version 8.0.

La base de données est hébergée sur les serveurs de Heroku

4 - PROCÉDURE DE DÉPLOIEMENT

4.1 - Déploiement de l'application web

4.1.1 - Composition de l'application web

L'application architectAPP est construite sous la forme d'une archive ZIP contenant les répertoires suivants:

architectAPP : Contient les fichiers de configuration de l'application et de Django.

Procfile : Contient les instructions pour démarrer l'application

requirements.txt : contient les librairies nécessaires pour que l'application fonctionne.

Static : Contient les fichiers .CSS et .JS de l'application.

Templates : Contient les fichiers HTML de l'application.

Ventes : Contient les fichiers liés au package « Ventes » de l'application.

Production : Contient les fichiers liés au package « Production » de l'application.

Docs : Contient la documentation de l'application.

Le fichier **Procfile** contient les instructions exécutées par heroku pour démarrer l'application. Le contenu de ce fichier doit a minima être le suivant pour que l'application fonctionne :

web: unicorn architectapp.run :app

4.1.3 - Configuration

Voici les différents fichiers de configuration :

❑ **procfile** : fichier contenant les commandes utilisées par heroku pour le déploiement de l'application.

❑ **requirements.txt** : fichier contenant la liste des librairies à installer sur le serveur heroku pour que l'application fonctionne correctement.

4.1.4 - Déploiement

Heroku propose différentes méthodes pour déployer une application. Nous conseillons cependant d'effectuer un déploiement en ligne de commande en utilisant « Heroku CLI ». Les étapes présentées ci-dessous nécessite que « heroku CLI » soit installé, que le projet soit suivi avec « Git » et il faut être sur la branche « master » du projet.

4.1.4.1 - Création de l'application :

Une fois « heroku cli » installé la connexion au compte Architect effectuée, il faut se rendre à la racine du projet de l'application et entrer la commande suivante :

{{Entreprise}}
<siteWebEntreprise>

<AdresseEntreprise> – <Téléphone> – <Email>

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Xxxx – SIREN 999 999 999 – Code APE : 6202A

```
$ heroku create architectapp
```

4.1.4.3 - Envoyer l'application sur le serveur

Une fois les étapes précédentes réalisées, il suffit d'effectuer un « Push » sur le serveur :

```
$ git push heroku master
```

4.1.5 - Vérifications

Une fois le déploiement terminé, les lignes suivantes doivent apparaître dans le terminal :

```
remote : https:// architectapp.fr/ deployed to Heroku
```

```
remote: Verifying deploy... done.
```

Pour s'assurer du bon déploiement, il convient ensuite de se connecter à l'adresse www.architectpp.fr afin de vérifier que l'application est bien en ligne.

4.1.4.3 - Envoyer l'application sur le serveur

Une fois les étapes précédentes réalisées, il suffit d'effectuer un « Push » sur le serveur :

```
$ git push heroku master
```

4.1.5 - Vérifications

Une fois le déploiement terminé, les lignes suivantes doivent apparaître dans le terminal :

```
remote : https:// architectapp.fr/ deployed to Heroku
```

```
remote: Verifying deploy... done.
```

Pour s'assurer du bon déploiement, il convient ensuite de se connecter à l'adresse www.architectapp.fr afin de vérifier que l'application est bien en ligne.

4.2 - Déploiement de la base de données

Le déploiement de la base de données sur la plateforme Heroku nécessite que l'application ait été déployée comme présenté précédemment.



De la même manière que pour le déploiement de l'application, celui de la base de données est présenté ici en utilisant la ligne de commande et « heroku CLI ».


4.2.1 - Lancement des migrations

Avant d'importer les données en elle-même, il est nécessaire d'importer add-on postgre

\$ heroku addons:create heroku-postgresql

Se rendre dans datastores -> setting

 Datastores >  postgresql-elliptical-99955

SERVICE heroku-postgresql PLAN hobby-dev BILLING APP  pizzaappx

Overview Durability **Settings** Dataclips

ADMINISTRATION

Database Credentials

Get credentials for manual connections to this database.

Please note that **these credentials are not permanent**.

Heroku rotates credentials periodically and updates applications where this database is attached.

Host	ec2-54[REDACTED]-214.eu-west-1.compute.amazonaws.com
Database	d7dj7h2ns[REDACTED]
User	seqbzhdcfdmyo[REDACTED]
Port	5432
Password	662ba661[REDACTED]3ed9ffa1f43c348a764e95
URI	postgres://seqbzhdcfdmyo:662ba661[REDACTED]4840193f6930909b7def725b06a3a0[REDACTED]
Heroku CLI	heroku pg:psql postgresql-elliptical-99955 --app pizzaappx

Ensuite on accède à pgadmin et on se connecte au serveur grâce au information précédente

The screenshot shows the 'Create - Server' dialog box in pgAdmin. The 'General' tab is active. The 'Host name/address' field is empty and has a red border with a warning icon. The 'Port' field contains '5432'. The 'Maintenance database' field contains 'postgres'. The 'Username' field contains 'postgres'. The 'Password' field is empty. The 'Save password?' checkbox is unchecked. The 'Role' and 'Service' fields are empty. A red error message at the bottom says 'Name must be specified.' The bottom bar has 'Save', 'Cancel', and 'Reset' buttons.

4.2.3 - Import des données

Il est possible d'importer les données de la version de développement de l'application sur le serveur de production. Pour cela, il suffit d'exécuter le script SQL depuis pgAdmin.

4.2.4 - Vérifications

La vérification de l'état de la base de données peut se faire directement via le site de Heroku dans la rubrique « Ressources » du projet.

Il faut ensuite se rendre dans la partie « Add-ons » puis cliquer sur « Heroku Postgres :: Database ». On retrouve dans cet espace les informations sur la base de données

Ou en se connectant à pgAdmin nous pouvons vérifier aussi la base de donnée.

5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

La gestion de l'application peut se faire de deux manières, soit depuis le site de Heroku soit en ligne de commande après avoir téléchargé l'interface « heroku-cli » :

5.1 - Démarrage / arrêt sur le site heroku.com

Une fois déployée sur Heroku, l'application est par défaut en ligne.

Afin de désactiver l'application il faut se rendre dans la rubrique « Settings », puis changer le statut « Maintenance Mode » à « ON ».

Il suffira ensuite de faire la démarche inverse afin de rétablir l'application.

5.2 - Démarrage / arrêt en ligne de commande

Afin de démarrer ou arrêter l'application via la ligne de commande, il faut utiliser les commandes suivantes :

Mise en marche de l'application :

\$ heroku run

Arrêt de l'application :

\$ heroku ps :scale web=0

6 - PROCÉDURE DE MISE À JOUR

La mise à jour de l'application nécessite de passer celle-ci en mode maintenance. La mise en mode maintenance peut s'effectuer directement via le tableau de bord heroku (comme présenté dans la section précédente), soit directement en ligne de commande via la commande suivante :

\$ heroku maintenance :on

Lorsque le mode maintenance est activé, un message s'affiche lors de la connexion à l'application et les utilisateurs n'ont plus accès à celle-ci.

Une fois la mise à jour de l'application effectuée, l'application peut être réactivée comme ceci :

\$ heroku maintenance :off

7 - SUPERVISION/MONITORING

7.1 - Supervision de l'application web

Afin de tester que l'application web est toujours fonctionnelles, faire ceci...

La plateforme Heroku fournit des outils de monitoring dans la rubrique « Metrics » de l'application.

Plus d'informations sont disponibles dans la documentation de Heroku, à l'adresse suivante : <https://devcenter.heroku.com/categories/monitoring-metrics>

Nos recommandations de suivi à mettre en place sont a minima les suivantes :

- Utilisation/Saturation du processeur
- Utilisation/Saturation de la mémoire vive
- Rapprochement de la limite de stockage de la base de données.

8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

Le processus de sauvegarde et de restauration de l'application passe essentiellement par la sauvegarde des données de celle-ci. Heroku propose cette fonctionnalité. 8.1 - Sauvegarde de la base de données

\$ heroku pg:backups:capture --app architectapp 8.2 - Restauration de la base de données

\$ heroku pg:backups:restore <nom sauvegarde> DATABASE_URL --app architectapp

9 - GLOSSAIRE
