

Déploiement du projet Arkadia Aoo avec heroku

Prérequis

1. **Heroku CLI:** Assurez-vous d'avoir installé Heroku CLI. Si ce n'est pas le cas, suivez les instructions sur Heroku Dev Center.
2. **Compte Heroku:** Créez un compte sur [Heroku](https://heroku.com).
3. **GitHub Repository:** Ayez votre code source PHP prêt et hébergé sur un repository GitHub.
4. **Composer:** Assurez-vous que votre projet PHP utilise Composer pour la gestion des dépendances.

```
C:\Windows\system32>npm install -g heroku
npm warn EBADENGINE Unsupported engine {
npm warn EBADENGINE   package: 'heroku@8.11.5',
npm warn EBADENGINE   required: { node: '>=16.20.0' },
npm warn EBADENGINE   current: { node: 'v20.12.2', npm: '10.7.0' }
npm warn EBADENGINE }
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated osenv@0.1.5: This package is no longer supported.
npm warn deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm warn deprecated @oclif/color@0.1.2: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/help@1.0.15: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/parser@3.8.17: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/errors@1.3.6: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/screen@3.0.8: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/config@1.18.17: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/command@1.8.36: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated strip-eof@2.0.0: Renamed to 'strip-final-newline' to better represent its functionality.
npm warn deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported
npm warn deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported
npm warn deprecated @oclif/config@1.18.16: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/screen@3.0.8: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @oclif/screen@1.0.4: Deprecated in favor of @oclif/core
npm warn deprecated uuid@3.3.2: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic
. See https://v8.dev/blog/math-random for details.
npm warn deprecated @oclif/screen@1.0.4: Deprecated in favor of @oclif/core
npm warn deprecated heroku-cli-util@8.0.12: This package has been deprecated. Install the new supported version at https://www.npmjs.com/package/@heroku/heroku-cli-util
npm warn deprecated cli-ux@6.0.9: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce asy
nc requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated @heroku-cli/plugin-ps@8.1.7: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated cli-ux@4.9.3: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated @heroku-cli/plugin-run@8.1.4: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.

added 831 packages in 1m
50 packages are looking for funding
```

Étape 1: Configuration Initiale de l'Application sur Heroku

1. Créer une Nouvelle Application Heroku

Ouvrir le terminal et exécutez la commande suivante pour créer une nouvelle application sur Heroku :

```
C:\Users\User1>heroku create arkadia-zoo
Creating  arkadia-zoo... !
! To create an app, verify your account by adding payment information. Verify now at https://heroku.com/verify Learn more at
! https://devcenter.heroku.com/articles/account-verification
```

2. Ajouter le Buildpack PHP

Configurer Heroku pour utiliser le buildpack PHP : créer un fichier `.buildpack` à la racine du projet avec le contenu suivant :

```
https://github.com/heroku/heroku-buildpack-php.git
```

Étape 2: Configurer les Add-ons pour MySQL et MongoDB

1. Ajouter jawsdb MySQL

jawsdb est un add-on populaire pour MySQL sur Heroku :

```
C:\Users\User1>heroku addons:create jawsdb:kitefin -a arkadia-zoo
Creating jawsdb:kitefin on arkadia-zoo... free
Resource is being provisioned.
jawsdb-rugged-07668 is being created in the background. The app will restart when complete...
Use heroku addons:info jawsdb-rugged-07668 to check creation progress
Use heroku addons:docs jawsdb to view documentation
```

Interface heroku :

Personal > arkadia-zoo ☆ Open app More ▾

GitHub bouzelmat/applicationArkadia main

Overview Resources Deploy Metrics Activity Access Settings

Basic Dynos Change Dyno Type

web vendor/bin/heroku-php-apache2 ~\$0.010/hour

Add-ons Find more add-ons

Quickly add add-ons from Elements

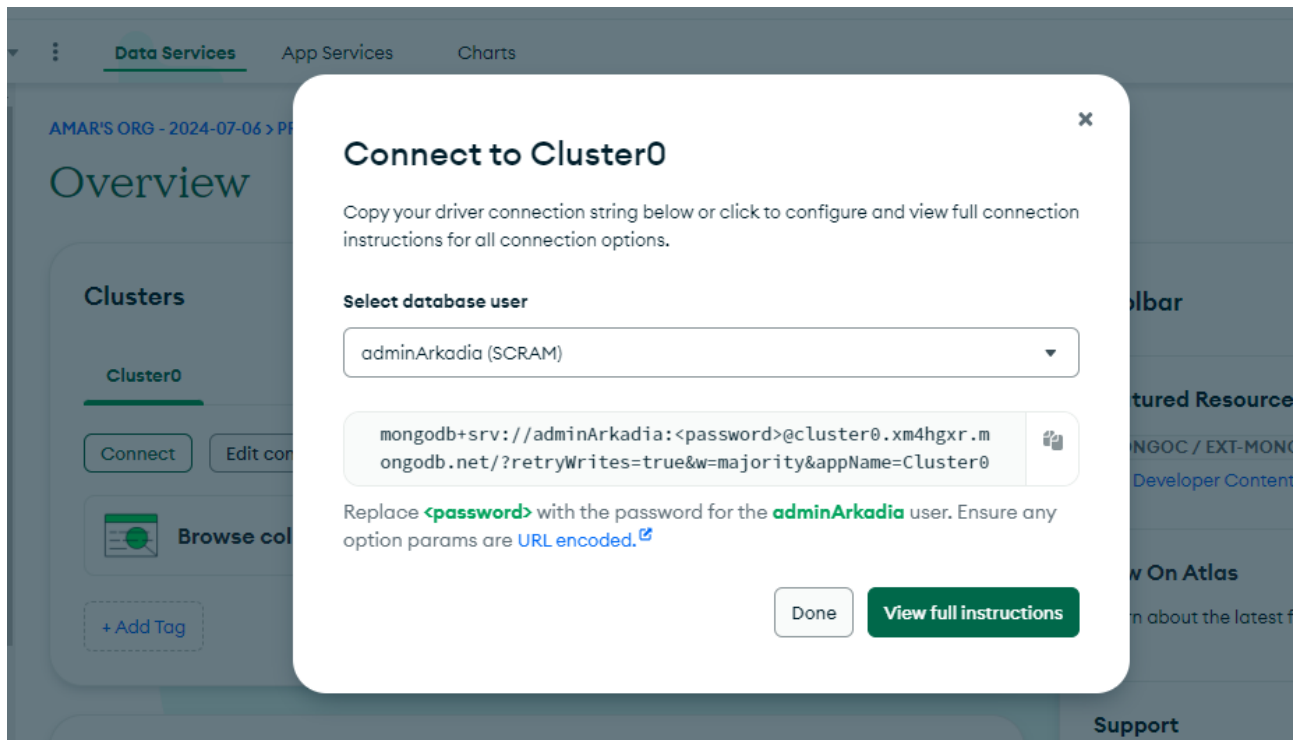
JawsDB MySQL Attached as JAWSDB Kitefin Shared Free

2. Ajouter MongoDB Atlas

Après avoir créer un compte il faut créer une nouvelle organisation et un nouveau projet pour héberger le cluster mongoDB .

Récupérer la chaîne de connexion du cluster Atlas en cliquant sur « Connecter » sur le cluster.

Remplacez le mot de passe dans la chaîne de connexion par celui de l'utilisateur créé précédemment



mettre à jour la connexion à MongoDB pour utiliser la variable d'environnement :

```
k>php
namespace ecf_arkadia\Config;

use MongoDB\Client;

class MongoDBConnection {
    private $uri;
    private $client;

    public function __construct() {
        $this->uri = getenv('MONGODB_URI') ?: (require __DIR__ . '/mongo_config.php')['mongo_uri'];
        if (!$this->uri) {
            throw new \Exception("L'URI MongoDB n'est pas définie");
        }
    }

    public function connect() {
        if ($this->client === null) {
            try {
                $quotaguardUrl = getenv('QUOTAGUARDSTATIC_URL');
                $proxy = parse_url($quotaguardUrl);

                $options = [
                    'tls' => true,
```

sur l'interface heroku aller a la section « Config Vars » et cliquer sur le bouton « Reveal Config Vars ». puis Cliquer sur « Ajouter » pour ajouter une nouvelle variable d'environnement . Pour le nom , entrer MONGODB_URI . et nous ajoutons la clé a cette variable.

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some additions come with their own.

Config Vars

JAWSDB_URL	mysql://ljabbiyob2py5uq:j41l
MONGODB_URI	mongodb+srv://adminArkadia:po
QUOTAGUARDSTATIC_URL	http://tlfuynkx43qg0z:vtx10k:
KEY	VALUE

Etape 3 : Configuration de l'Application PHP pour Heroku

1.Fichier composer . json

Il faut s'assurer que le projet PHP a un fichier composer . json correctement configuré pour gérer les dépendances.

```
"name": "ecf/ecf_arkadia",
"description": "premier projet",
"type": "project",
"license": "MIT",
"autoload": {
    "psr-4": {
        "ecf_arkadia\\": "src/"
    }
},
"authors": [
    {
        "name": "bouzelmat",
        "email": "omarbouzelmat99@gmail.com"
    }
],
"require": {
    "php": "^8.2",
    "ext-mysqli": "*",
    "ext-mongodb": "*",
    "ext-openssl": "*",
    "phpmailer/phpmailer": "^6.9",
    "mongodb/mongodb": "^1.15",
    "ext-curl": "*"
},
```

2. Fichier Procfile

Créer un fichier `Procfile` à la racine du projet pour indiquer à Heroku comment exécuter l'application.

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

3. Configurer la Connexion à la Base de Données

Dans l'application PHP, configurer la connexion à MySQL et MongoDB en utilisant les variables d'environnement définies plus tôt :

```
<?php
namespace ecf_arkadia\Config;

class Database {
    private $host;
    private $port;
    private $db_name;
    private $username;
    private $password;
    public $conn;

    public function __construct() {
        if (getenv('JAWSDB_URL')) {
            $url = parse_url(getenv('JAWSDB_URL'));
            $this->host = $url['host'] ?? null;
            $this->port = $url['port'] ?? null;
            $this->db_name = isset($url['path']) ? ltrim($url['path'], '/') : null;
            $this->username = $url['user'] ?? null;
            $this->password = $url['pass'] ?? null;
        } else {
            $config = require __DIR__ . '/config.php';
            $this->host = $config['db_host'];
            $this->port = $config['db_port'];
            $this->db_name = $config['db_name'];
            $this->username = $config['db_username'];
            $this->password = $config['db_password'];
        }

        if (!$this->host || !$this->username || !$this->password || !$this->db_name) {
            throw new \Exception("Configuration de base de données incomplète");
        }
    }

    public function getConnection() {
        $this->conn = null;
    }
}
```

Étape 6: Déployer l'Application


Allez sur le tableau de bord Heroku, sélectionnez l'application et aller dans l'onglet "Deploy". Sous "Deployment method", choisir "GitHub" et connecter le compte GitHub si ce n'est pas déjà fait. Ensuite recherchez le repository du projet.


- Configurer le Déploiement Automatique (optionnel)

Pour ce qui me concerne j'ai configuré le déploiement automatique pour chaque push sur la branche principale. Pour ce type de déploiement il faut aller sous "Automatic deploys", puis cliquer sur "Enable Automatic Deploys".

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

 You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

 Automatic deploys from `main` are enabled

Every push to `main` will deploy a new version of this app. **Deploys happen automatically**: be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. [Learn more](#).

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Disable Automatic Deploys


Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more](#).

Choose a branch to deploy

 main

Deploy Branch