

Documentation technique Arcadia

Réflexion initiale

Pour ce projet Arcadia j'ai décidé d'utiliser le Framework Symfony car c'est le Framework sur lequel j'ai appris pendant ma formation. Et il a l'avantage de pouvoir gérer facilement l'aspect admin que ce soit la création de CRUD ou bien son Dashboard et même tout l'aspect sécurité grâce à ses nombreux bundles (Back-end). Et j'ai pu réaliser ma partie Front-end en même temps grâce au template Twig.

Environnement de travail

Environnement :

- PHP >= 8.1
- MYSQL >= 5.7 (WAMP)
- Symfony 5.8.16
- Bootstrap 5.3
- Easyadmin 4.10 VSCode

Extensions VSCODE:

- PHP Intelephense
- Twig Language 2
- DotENV
- phpfmt PHP formatter
- Material Icon Theme
- PHP Getter & Setter
- PHP Namespace Resolver
- YAML
- PHPDoc Comment Prettier - Code formatter

Installation du projet Arcadia :

Avoir défini php dans les variables d'environnement et avoir Xdebug d'activé.

Installer composer

Installer symfony cli

Vérifier les requirements :

`symfony check:requirement`

Créer un nouveau projet symfony

`symfony new my_project --webapp`

Relancer l'indexation dans VSCODE : CTRL+MAJ+P -> Intelephense: index

Lancer un serveur symfony `server:start`

Installer webpack Encore 2.1:

`composer require symfony/webpack-encore-bundle`

`npm install`

Pour compiler en dev

`npm run dev`

Pour compiler en prod

```
npm run build
```

Pour compiler en temps réel les changements

```
npm run watch
```

Webpack Encore : SASS et Bootstrap:

```
npm install sass-loader sass --save-dev
```

Modifier webpack.config.js et ajouter .enableSassLoader()

```
npm install bootstrap --save-dev
```

Webpack Encore : Images

```
npm install file-loader --save-dev
```

Modifier webpack.config.js et ajouter

```
.copyFiles({  
  from: './assets/images',  
  to: 'images/[path][name].[ext]',  
})
```

Easyadmin:

Installation du bundle easyadmin

```
composer require easycorp/easyadmin-bundle
```

Installation admin dashboard

```
php bin/console make:admin:dashboard
```

Vich/uploader-bundle

Installation vich/uploader composer require vich/uploader-bundle

Modifier le fichier config/packages/vich_uploader.yaml mappings: zoo: uri_prefix:
/images/nom_dossier upload_destination: '%kernel.project_dir%/public/images/zoo' namer:
Vich\UploaderBundle\Naming\SmartUniqueNamer

Modifier l'entité

```
use Vich\UploaderBundle\Mapping\Annotation as Vich;
```

```
#[Vich\Uploadable]
```

```
#[Vich\UploadableField(mapping: 'nom_dossier', fileNameProperty: 'imageName', size:  
'imageSize')] private ?File $imageFile = null;
```

```
#[ORM\Column(nullable: true)]
```

```
private ?string $imageName = null;
```

```
#[ORM\Column(nullable: true)]
```

```
private ?int $imageSize = null;
```

```
#[ORM\Column(nullable: true)]
```

```
private ?\DateTimeImmutable $updatedAt = null;
```

```
/**
```

```
 * If manually uploading a file (i.e. not using Symfony Form) ensure an instance
```

```
 * of 'UploadedFile' is injected into this setter to trigger the update. If this
```

```
 * bundle's configuration parameter 'inject_on_load' is set to 'true' this setter
```

```
 * must be able to accept an instance of 'File' as the bundle will inject one here
```

```
 * during Doctrine hydration.
```

```
 *
```

```
 * @param File|\Symfony\Component\HttpFoundation\File\UploadedFile|null $imageFile
```

```
 */
```

```
public function setImageFile(?File $imageFile = null): void
```

```
{
```

```
    $this->imageFile = $imageFile;
```

```

if (null !== $imageFile) {

    // It is required that at least one field changes if you are using doctrine

    // otherwise the event listeners won't be called and the file is lost

    $this->updatedAt = new \DateTimeImmutable();

}

}

```

```

public function getImageFile(): ?File

{

    return $this->imageFile;

}

```

```

public function setImageName(?string $imageName): void

{

    $this->imageName = $imageName;

}

```

```

public function getImageName(): ?string

{

    return $this->imageName;

}

```

```

public function setImageSize(?int $imageSize): void

```

```
{  
    $this->imageSize = $imageSize;  
}  
  
public function getImageSize(): ?int  
{  
    return $this->imageSize;  
}
```


Modèle conceptuel de donnée

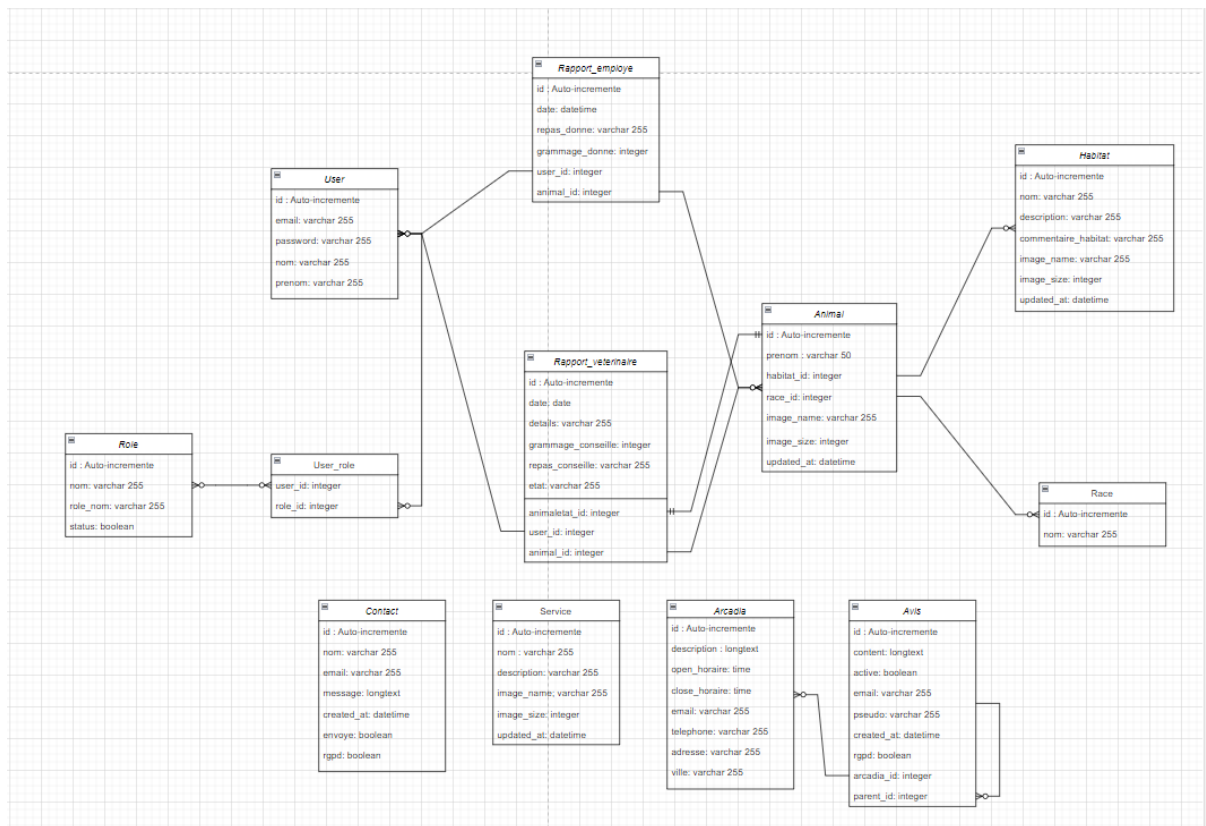
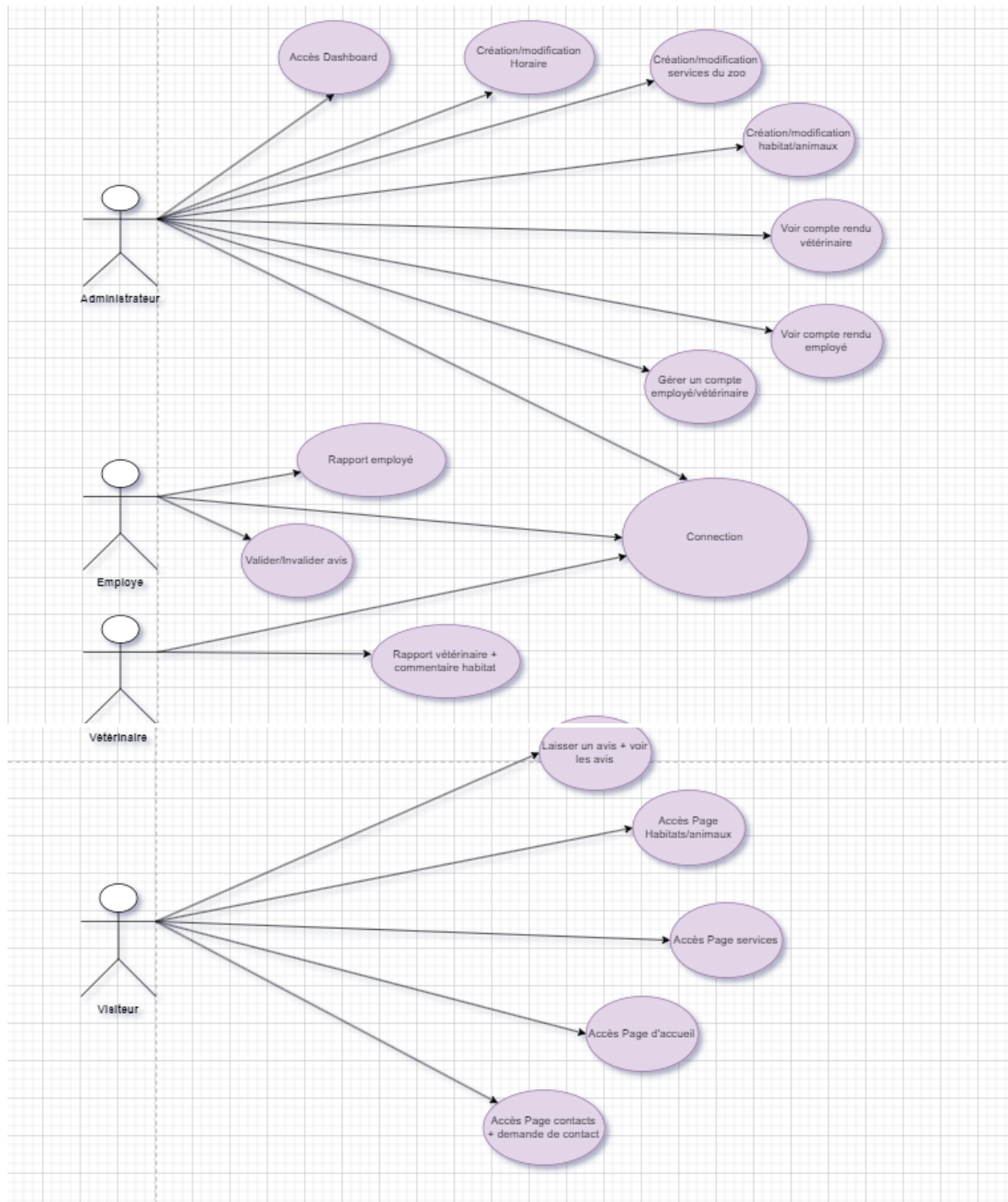


Diagramme de cas d'utilisation




Déploiement

Pour le déploiement j'ai fait mon déploiement avec Heroku.



Après avoir crée mon compte et m'être connecté j'ai pu crée mon application sur heroku.

App name



arcadia-ecf-stephen1 is available

Choose a region

 Europe 

Add to pipeline...

Create app Cancel


J'ai lié mon compte github à mon application créée sur le site d'heroku.

Installé une base de données

Heroku Platform


HEROKU

Online Order Form


 Provision this add-on to an app
JawsDB MySQL
[View on the Elements Marketplace](#)

Add-on plan

Kitefn Shared - Free

 JawsDB MySQL
Kitefn Shared - Free

→

 arcadia-ecf-stephen
Personal apps

In summary this will provision JawsDB MySQL on your personal arcadia-ecf-stephen application, using the Kitefn Shared plan (Free).

[Change destination app](#)

By submitting this order form, you agree that the Add-on is governed by the applicable provider's terms of use, and the Heroku Services are governed by the [Salesforce Master Subscription Agreement](#) unless (except for free customers) you have entered into a written Master Subscription Agreement executed by SFDG for the Heroku Services as referenced in the Documentation.









Submit Order Form

eu.com Blogs Careers Documentation **Support**

Terms of Service Privacy Cookies © 2024 Salesforce

Configurer mes variables

Config Vars Hide Config Vars

APP_ENV	prod	 
APP_SECRET	270fc8dc874ddf31370632e771d8798d	 
DATABASE_URL	mysql://ia7tms7jvbidltns:s4p37u5pf0qyn7hs	 
JAWSDB_URL	mysql://ia7tms7jvbidltns:s4p37u5pf0qyn7hs	 
KEY	VALUE	Add

J'ai configuré ces mêmes variables dans mon fichier .env de mon projet.

```
12 # https://symfony.com/doc/current/configuration/secrets.html
13 #
14 # Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.2).
15 # https://symfony.com/doc/current/best_practices.html#use-environment-variables-for-infrastructure-configuration
16
17 ###> symfony/framework-bundle ###
18 APP_ENV=prod
19 APP_SECRET=270fc8dc874ddf31370632e771d8798d
20 ###< symfony/framework-bundle ###
21
22 ###> doctrine/doctrine-bundle ###
23 # Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url
24 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
25 #
26 # DATABASE_URL="sqlite://%kernel.project_dir%/var/data.db"
27 # DATABASE_URL="mysql://app:!ChangeMe!@127.0.0.1:3306/app?serverVersion=8.0.32&charset=utf8mb4"
28 # DATABASE_URL="mysql://app:!ChangeMe!@127.0.0.1:3306/app?serverVersion=10.11.2-MariaDB&charset=utf8mb4"
29 # DATABASE_URL="postgresql://app:!ChangeMe!@127.0.0.1:5432/app?serverVersion=16&charset=utf8"
30 DATABASE_URL="mysql://ia7tms7jvbidltns:s4p37u5pf0qyn7hs@1t80glfe2gj8p5n2.chr7pe7iyngqr.eu-west-1-rds.amazonaws.com:3306/er18hvv6n0by6q6i"
31 ###< doctrine/doctrine-bundle ###
32
33 ###> symfony/messenger ###
34 # Choose one of the transports below
35 # MESSENGER_TRANSPORT_DSN=amqp://guest:guest@localhost:5672/?xzf/messages
36 # MESSENGER_TRANSPORT_DSN=redis://localhost:6379/messages
37 MESSENGER_TRANSPORT_DSN=doctrine://default?auto_setup=0
```

J'ai créé ensuite un fichier .htaccess avec la commande :

Composer require symfony/apache-pack.

```
steph@DESKTOP-7350600 MINGW64 /d/Projet/arcadia_ecf -deployment (main)
$ composer require symfony/apache-pack
./composer.json has been updated
Running composer update symfony/apache-pack
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking symfony/apache-pack (v1.0.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing symfony/apache-pack (v1.0.1): Extracting archive
Generating autoload files
116 packages you are using are looking for funding.
Use the "composer fund" command to find out more!

Symfony operations: 1 recipe (a312addcd464d1f17774f3bcdcf8f39e)
- [WARNING] symfony/apache-pack (>=1.0): From github.com/symfony/recipes-contrib:main
  The recipe for this package comes from the "contrib" repository, which is open to community contributions.
  Review the recipe at https://github.com/symfony/recipes-contrib/tree/main/symfony/apache-pack/1.0

Do you want to execute this recipe?
[y] Yes
[n] No
[a] Yes for all packages, only for the current installation session
[p] Yes permanently, never ask again for this project
(defaults to n): y
- Configuring symfony/apache-pack (>=1.0): From github.com/symfony/recipes-contrib:main
Executing script cache:clear
```

Puis le fichier Procfile.

```
Procfile  X
Procfile > web
1  release: php bin/console cache:clear && php bin/console cache:warmup
2  web: vendor/bin/heroku-php-apache2 public/ && yarn install && yarn encore production
```

Une fois ce processus effectué j'ai fait une merge de mon dossier mis a jours sur la branch main de mon github.

J'ai lié mon compte github à mon application créée sur le site d'heroku.

Et j'ai pu faire la commande « git push heroku main » pour push mon repository git sur heroku.

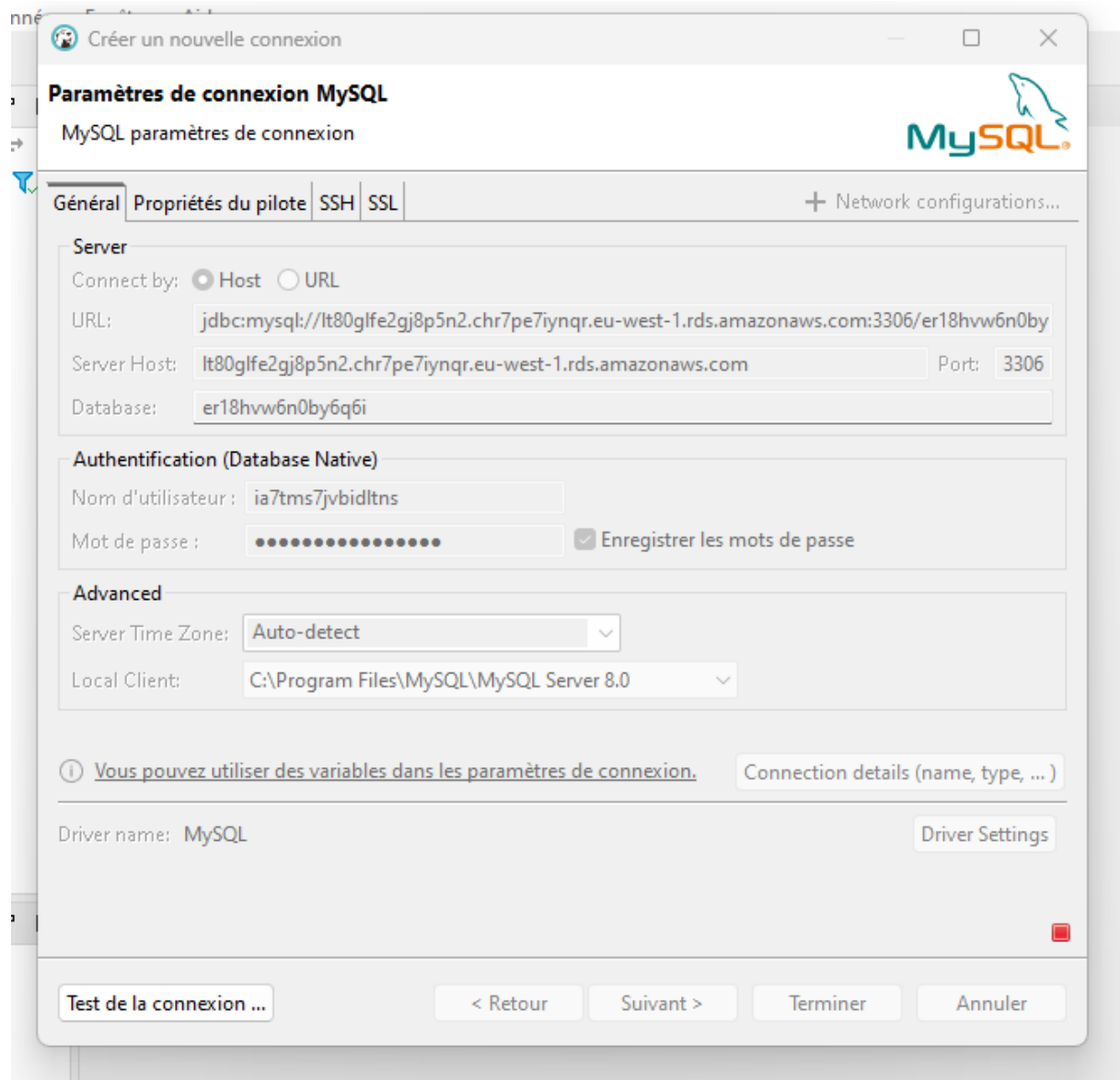
```
steph@DESKTOP-73506Q0 MINGW64 /d/Projet/arcadia_ecf_deployment (main)
$ git push heroku main
Enumerating objects: 409, done.
Counting objects: 100% (409/409), done.
Delta compression using up to 12 threads
Compressing objects: 100% (380/380), done.
Writing objects: 100% (409/409), 4.86 MiB | 1.69 MiB/s, done.
Total 409 (delta 194), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (194/194), done.
remote: Updated 160 paths from 2cbb07a
remote: Compressing source files... done.
remote: Building source:
remote:
remote: ----> Building on the Heroku-22 stack
remote: ----> Determining which buildpack to use for this app
remote: !    Warning: Multiple default buildpacks reported the ability to handle this app. The first buildpack in the list below will be used.
remote:           Detected buildpacks: PHP,Node.js
remote:           See https://devcenter.heroku.com/articles/buildpacks#buildpack-detect-order
remote: ----> PHP app detected
remote: ----> Bootstrapping...
remote: ----> Preparing platform package installation...
remote: ----> Installing platform packages...
remote:       - php (8.3.5)
remote:       - apache (2.4.59)
remote:       - composer (2.7.2)
remote:       - nginx (1.24.0)
remote: ----> Installing dependencies...
remote:       Composer version 2.7.2 2024-03-11 17:12:18
remote:       Installing dependencies from lock file
remote:       Verifying lock file contents can be installed on current platform.
remote:       Package operations: 101 installs, 0 updates, 0 removals
remote:       - Downloading symfony/flex (v2.4.5)
remote:       - Downloading symfony/runtime (v7.0.7)
remote:       - Downloading doctrine/cache (2.2.0)
remote:
remote: // Warming up the cache for the prod environment with debug false
remote: [OK] Cache for the "prod" environment (debug=false) was successfully warmed.
remote: Waiting for release.... done.
To https://git.heroku.com/arcadia-ecf-stephen.git
* [new branch]    main -> main

steph@DESKTOP-73506Q0 MINGW64 /d/Projet/arcadia_ecf_deployment (main)
$
```

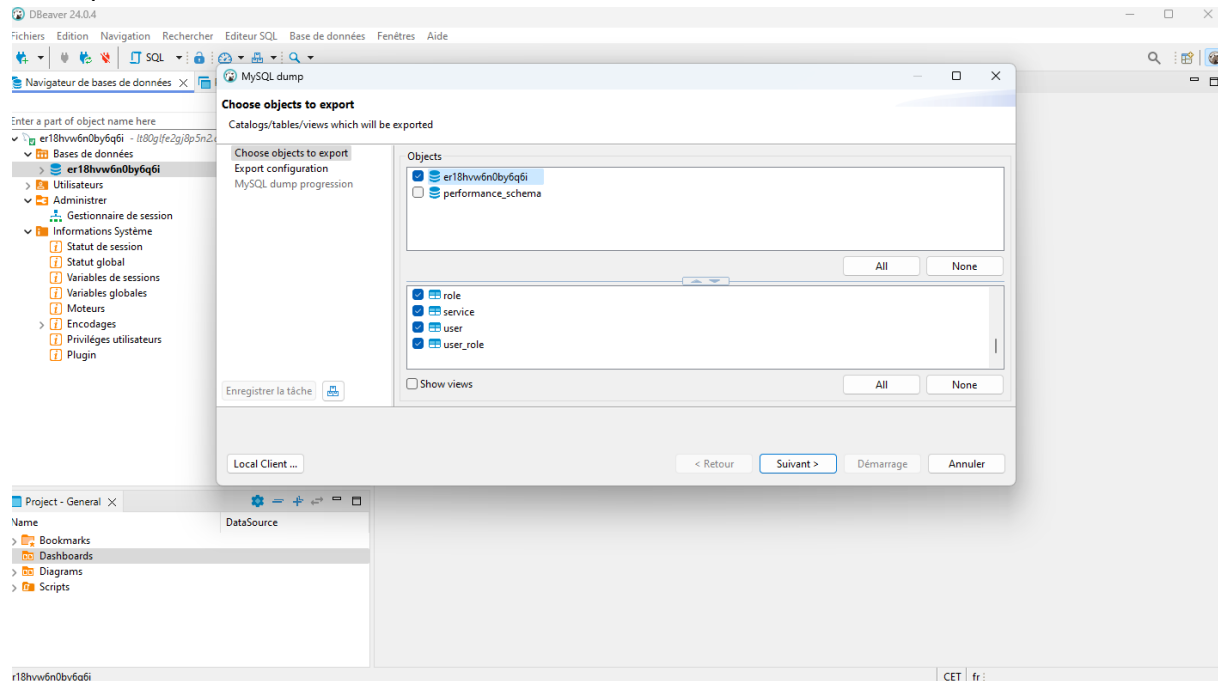
Mon application est en ligne !

Pour gérer la base de données distante afin de pouvoir créer un administrateur j'ai utilisé le logiciel DBeaver.

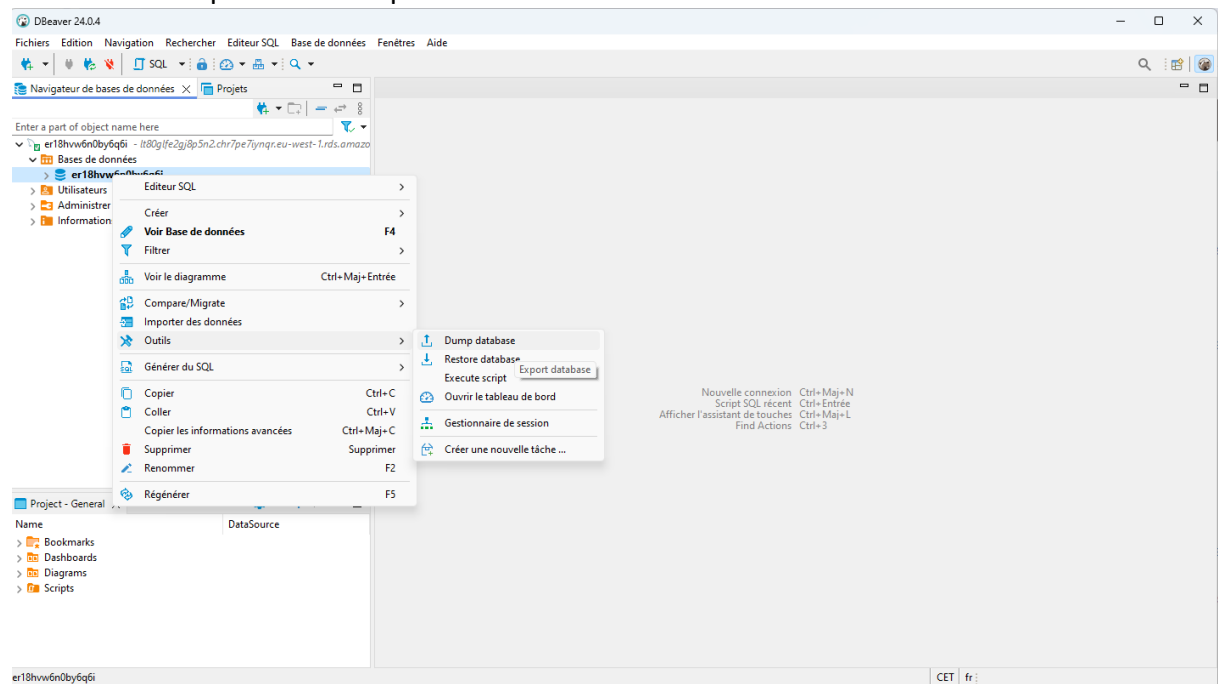
Je me suis connecté à la base grâce aux informations récupéré sur heroku.



J'ai exporté ma base de données



J'ai fait un dump de ma base pour avoir mes tables.



Puis j'ai pu intégrer toutes les données.

