Le déploiement

1. Le nom de domaine, l’hébergeur et ses services
2. La configuration des services (bases de données et mail)
3. La création des secrets de production
4. La création des .htaccess
5. L’installation du projet en production

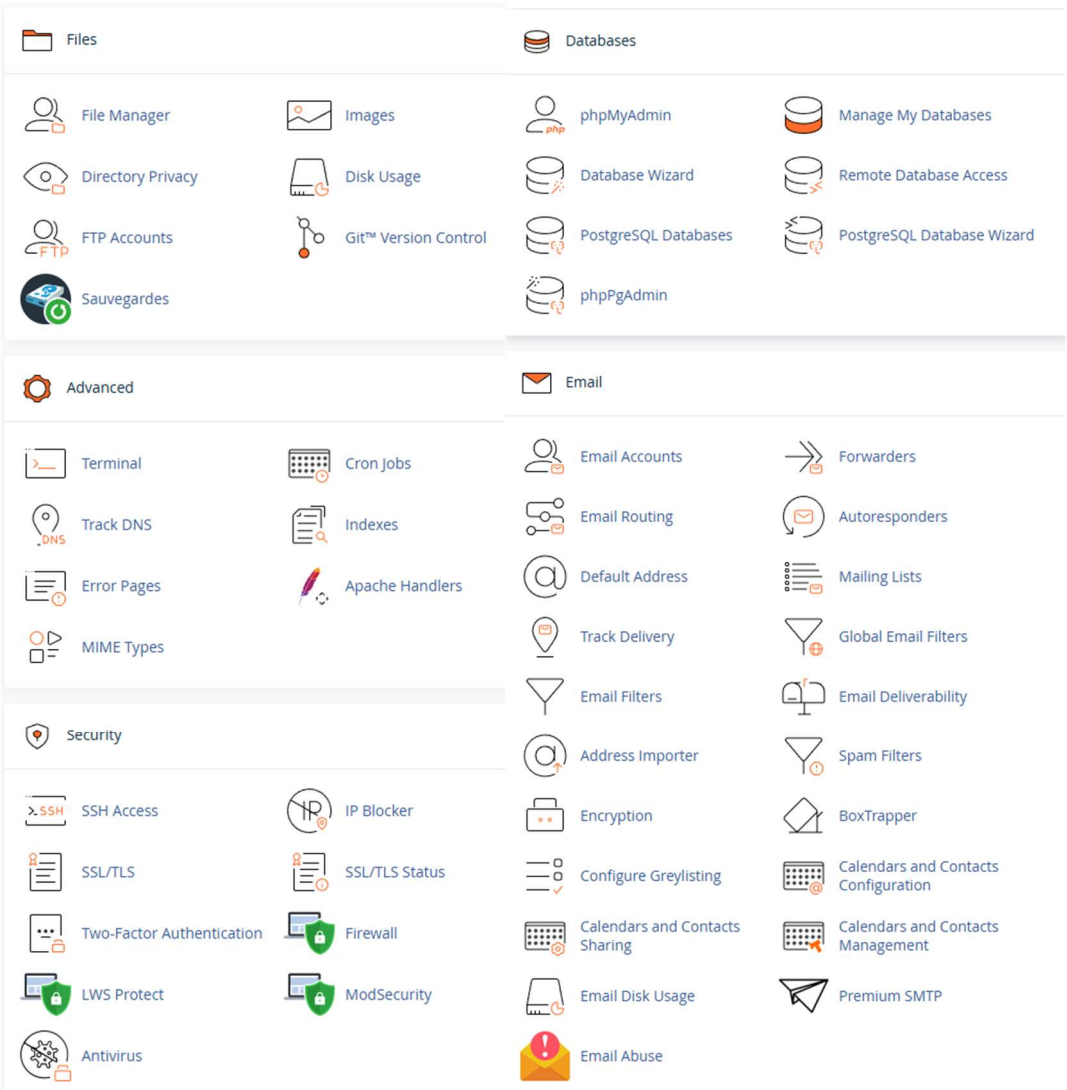
1. Le nom de domaine, l'hébergeur et ses services

La première étape consiste à acheter le nom de domaine (ecoride.site). Cela permet d'accéder à une interface de gestion du domaine, notamment pour configurer les DNS.

Le projet étant déployé sur un hébergement CPanel en tant que « site compagnon », la prochaine étape est de reconfigurer les DNS du domaine pour qu'ils pointent vers l'hébergement principal choisi (cet hébergement utilise Apache comme serveur web).

Dès que la propagation DNS est terminée, on peut créer le certificat SSL attaché au domaine à l'aide de l'outil dédié.

L'interface CPanel offre plusieurs services essentiels, notamment :



2. La configurations des services

Pour connecter les services de bases de données (SQL et MongoDB) à l'application, nous avons besoin de plusieurs informations : un nom de base de données, un nom d'utilisateur, et un mot de passe pour chaque service.

a. MySQL

Via l'outil « Manage My Databases » de CPanel, on crée la base de données SQL et un utilisateur dédié à l'application auquel on accorde les droits nécessaires. Un mot de passe fort est requis pour sécuriser cet utilisateur.

b. MongoDB

Comme LWS ne fournit pas de support natif pour MongoDB, on crée donc un compte sur MongoDB Atlas. Là aussi, il est nécessaire de créer un utilisateur avec les droits appropriés et un mot de passe fort. De plus, nous devons autoriser l'adresse IP de l'application à accéder à la base de données (dans le menu « Security > Network Access »).

Nous devons également autoriser MongoDB Atlas à accéder à l'application via une connexion sortante.

c. Service mail

Pour le service mail, on crée une adresse dédiée à l'application(« mailer@ecoride.site ») en utilisant les outils fournis par CPanel. Là encore, un mot de passe fort doit être utilisé.

En production, on peut d'ores et déjà créer un cron job pour consommer les messages enregistrés en base de données (le script se termine et se relance toutes les minutes).
Le script du cron est le suivant :

```
/usr/local/bin/php /home/cp1765551p06/public_html/ecoride.site/bin/console messenger:consume --time-limit=55 --memory-limit=128M async > /home/cp1765551p06/public_html/ecoride.site/log.txt 2>&1
```

3. La création des secrets de production

Toutes les informations obtenues lors de la configuration des services sont utilisées pour configurer les secrets de production. Ceux-ci comprennent les variables suivantes : DATABASE_URL, MONGO_URL, et MAILER_DSN.

On commence par générer une paire de clés cryptographiques pour l'environnement de production avec la commande `php bin/console secrets :generate-keys --env=prod`.

Ensuite, on définit les secrets avec les commandes :

```
php bin/console secrets :set DATABASE_URL --env=prod
php bin/console secrets :set MONGODB_URL --env=prod
php bin/console secrets :set MAILER_DSN --env=prod
```

4. La création des .htaccess

Pour rappel : l'hébergement utilise Apache

a. Forcer le protocole HTTPS et rediriger les requêtes vers le dossier

À la racine du projet, on crée le fichier .htaccess suivant pour forcer le protocole HTTPS et rediriger les requêtes vers le dossier public/ du projet :

.htaccess

```
RewriteEngine on

RewriteCond %{HTTPS} !=on
RewriteCond %{HTTP:X-Forwarded-Proto} !https
RewriteCond %{HTTP_HOST} ^(?:www\.)?ecoride\.site(?:.*)$ [nc]
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [redirect=301,L]

RewriteCond %{REQUEST_URI} !^public
RewriteRule ^(.*)$ public/$1 [L]
```

b. Autoriser la réécriture des URL et pointer vers le fichier index.php

Dans le dossier public/ du projet, on crée un autre fichier .htaccess pour autoriser la réécriture d'URL et pointer vers le fichier index.php :

public > .htaccess

```
RewriteEngine On
RewriteBase /

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?$1 [L,QSA]
```

5. L’installation du projet en production

Pour rappel : le build des fichiers css et js est inclus dans le dépôt.

Après avoir configuré un tunnel SSH entre le dépôt GitHub et le serveur de production (en suivant un processus similaire à celui décrit dans la documentation de l'environnement de développement), on accède au terminal CPanel.

a. Cloner le dépôt distant

On se place dans le dossier cible (« public_html ») et on clone le repository distant avec la commande `git clone git@github.com:Lox-fr/ecoride.site.git` (le mot de passe de la clé SSH sera demandé pour effectuer cette opération). Les fichiers de la branche main sont maintenant disponibles sur le serveur de production.

b. Télécharger la clé privée

En utilisant le gestionnaire de fichiers de CPanel, on upload en direct la clé privée qui n’a pas été (et ne doit jamais être) commitée.

c. Exécuter le script de déploiement

Une fois la clé privée téléchargée, on retourne dans le terminal, on se place dans le répertoire du projet (cd ecoride.site), puis on exécute le script Composer personnalisé « firstDeploy » (`composer firstDeploy`).

Ce script a été créé en amont et contient :

```
"firstDeploy": [
    "composer install --no-dev --no-scripts --optimize-autoloader",
    "php bin/console secrets:decrypt-to-local --force --env=prod",
    "composer dump-env prod",
    "rm .env.prod.local",
    "rm config/secrets/prod/prod.decrypt.private.php",
    "php bin/console doctrine:database:drop --if-exists -f",
    "@sqlTablesCreation"
],
```

d. Supprimer la clé privée

Dès que l’exécution est terminée, on supprime la clé privée chargée manuellement.