

Sommaire :

- Cahier des charges
- Formalisme
- Stack
- Architecture du projet

Cahier des charges

- Gestion du conteneur
 - Possibilité de lancer des conteneurs
 - Possibilité de stopper des conteneurs
 - Possibilité d’attribuer des propriétés aux conteneurs
- Possibilité de visualiser les conteneurs en cours / arrêtés / en erreur
- Possibilité de supprimer un conteneur
- Possibilité de modifier un conteneur
- Possibilité de gérer les comptes / utilisateur pour le super admin
- Possibilité de modifier son compte
- Possibilité de visualiser une page de détaille sur le conteneur
- Possibilité d’ajouter / modifier des dockerfiles
- Possibilité de visualiser le log du conteneur

Formalisme

Pour ce projet, un formalisme des commits et des tickets sera essentiel afin de garantir un suivi de projet de qualité.

Pour cela, le formalisme suivant a été défini :

[FIX]	Définit une correction de bug dans un message de commit
[BUG]	Définit que le ticket représente un bug dans un ticket
[FEATURE]	Définit que le ticket représente une fonctionnalité ou que le commit représente l'ajout de fonctionnalité
[RELEASE]	Définit que le commit représente une release

[FIX]	Définit une correction de bug dans un message de commit
[DOC]	Définit que le ticket ou le commit représente un ajout ou une modification de documentation
[HOTFIX]	Définit que le ticket ou le commit a réalisé une correction de bug rapide et urgente sur une des branches de production (preprod / main)

Stack

- PHP
- Apache
- Twig
- SQL (maraiadb)
- Docker
- SSH

Architecture du projet

```

project-root/
├── assets/
│   ├── css/
│   │   └── *.css
│   ├── img/
│   │   └── *.png/jpg/jpeg
│   └── js/
│       └── *.js
├── cache/
├── docker/
│   ├── apache/
│   │   ├── certs/
│   │   ├── config/
│   │   └── logs/
│   ├── db/
│   ├── .gitignore
│   ├── apache.dockerfile
│   ├── composer.dockerfile
│   ├── dind.dockerfile
│   ├── php.dockerfile
│   └── test.dockerfile
├── documentation/
│   ├── BDD/model_entite_assoc.png
│   ├── maquette/
│   ├── CahierDesCharges.md
│   ├── UML_cas_utilisation.png
│   └── environnement/developpement.md
└── public/

```

```
|— sql/
|— src/
|— template/
|— tests/
|— .env
|— .gitignore
|— .gitlab-ci.yml
|— .htaccess
|— composer.json
|— composer.lock
|— dev.yaml
|— index.php
|— README.md
└— utils.yaml
```