

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2024
ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)	
Épreuve E5 - Conception et développement d'applications (option SLAM)	

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 1
Nom, prénom : REICHHART Damien		N° candidat : 02149931394
Épreuve ponctuelle <input type="checkbox"/>	Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : 28 / 04 / 2025
Organisation support de la réalisation professionnelle Du code exécuté sur le système d'exploitation d'un serveur.		
Intitulé de la réalisation professionnelle Atelier de professionnalisation 3		
Période de réalisation : Septembre 2024 – Janvier 2025 Modalité : <input type="checkbox"/> Seul(e) <input checked="" type="checkbox"/> En équipe		Lieu : Cci Campus Alsace Strasbourg
Compétences travaillées X Concevoir et développer une solution applicative X Assurer la maintenance corrective ou évolutive d'une solution applicative X Gérer les données		
Conditions de réalisation¹ (ressources fournies, résultats attendus) Lister, démarrer, arrêter, modifier et supprimer des conteneurs. Gérer les images Docker. Visualiser les logs et les statistiques des conteneurs. Les résultats attendus comprennent une application web intuitive, robuste et sécurisée, capable de communiquer efficacement avec l'API Docker, ainsi qu'une documentation technique détaillant l'architecture, l'installation et l'utilisation de la solution.		
Description des ressources documentaires, matérielles et logicielles utilisées² Ressources documentaires : <ul style="list-style-type: none"> • Cahier des charges et spécifications fonctionnelles fournies Ressources matérielles : <ul style="list-style-type: none"> • Poste de travail avec configuration adaptée pour le développement et les tests Ressources logicielles : <ul style="list-style-type: none"> • Environnement de développement intégré (EDI) : Visual Studio Code, PhpStorm • Langages et frameworks : <ul style="list-style-type: none"> ○ PHP ○ SQL ○ Twig • Technologies et outils : <ul style="list-style-type: none"> ○ Docker ○ Systèmes de gestion de base de données Mariadb ○ Outils de gestion de versions Git et hébergement sur GitLab ○ Serveur web pour déploiement Apache ○ Mermaid et PlantUML pour la documentation 		
Modalités d'accès aux productions³ et à leur documentation⁴ L'accès aux productions et documentations se font sur gitlab à l'url suivante : https://gitlab.com/DamienReichhart/CCI-BTS-SIO-23-25-Atelier-Professionnalisation-3 La documentation du projet est également accessible à cette url : https://damien-reichhart.fr/		

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

Table des matières

Cahier des charges fonctionnel.....	3
Formalisme	3
Gestion de projet.....	3
Réalisations.....	4
Architecture	5
Documentation technique	6
Diagrammes de classe	6
Diagramme de contexte statique.....	6
Diagramme de cas d'utilisation	7
Modèle entité association	7
Diagrammes de séquence	8
Diagramme de séquence des réseaux docker	8
Diagramme de séquence des requêtes http	9
Diagramme des volumes docker.....	9
Diagramme du lancement de l'application	10
Diagramme des couches de sécurité docker	11
Cycle de vie de la requête http.....	12
Intégration docker dans php	13
Diagramme des flux de données.....	14
Diagramme des étapes des middlewares	15
Diagramme de représentation du MVC.....	16
Captures d'écrans de l'application	17
Page de connexion.....	17
Gestion d'un conteneur.....	18
Gestion d'un serveur.....	19
Gestion des serveurs	20
Gestion des dockerfiles.....	21
Gestion des utilisateurs	23

Cahier des charges fonctionnel.

Une application web intuitive, robuste et sécurisée pour docker, capable de communiquer efficacement avec l'API Docker, ainsi qu'une documentation technique détaillant l'architecture, l'installation et l'utilisation de la solution.

Les besoins fonctionnels de la solution ont été définis comme tel :

- Gestion du conteneur
 - Possibilité de lancer des conteneurs
 - Possibilité de stopper des conteneurs
 - Possibilité de modifier/supprimer/ajouter des attribues 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 d'ajouter / modifier des dockerfiles personnalisés destinés à la solution
- Possibilité de visualiser le log du conteneur
- Possibilité d'ajouter un serveur à manager

Formalisme

Pour ce projet, un formalisme des commits et des tickets a été mis en place afin de garantir un suivi de projet de qualité.

Pour cela, le formalisme suivant a été défini avec les préfixes suivants :

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

Gestion de projet

Le projet a été mené selon la méthodologie Waterfall, ou cascade en français.

Dans ce cadre, l'ensemble des tâches et des étapes du projet ont été planifiées en amont, puis structurées sous forme de tickets et de jalons (Milestones).

Chaque membre de l'équipe s'est vu attribuer des tickets correspondant aux fonctionnalités à développer ou aux bugs à corriger. Cela permettait un suivi clair de l'avancement, une activité en cours était facilement identifiable grâce aux tickets assignés, assurant ainsi une bonne coordination au sein de l'équipe.

Réalisations

Dans la réalisation de ce projet, j'ai eu la chance de réaliser les éléments suivants :

- Création du schéma de la base de données.
- Gestion des interactions avec les autres serveurs via un client ssh.
- Gestion des interactions avec docker en temps réel dont la mise en place de l'architecture qui sera définie ci-dessous.
- Gestion des conteneurs :
 - Ajout / suppression / modification des conteneurs
 - Ajout / suppression / modification des attributs des conteneurs avec le log, stats, ports, volumes, environnement et commandes.
 - Gestion du statut marche / arrêt d'un conteneur
 - Gestion des dockerfiles personnalisés pour la solution, de l'étape de création / modification jusqu'à l'envoi et l'utilisation sur le serveur.
- Mise en place du système d'authentification
- Amélioration de la superclasse Model pour permettre des interactions plus simples avec la base de données
- Rédaction de la documentation liée aux diagrammes de classe, du readme et de la documentation sur l'environnement de développement.
- Réalisation de l'intégralité des templates twig
- Réalisations de scripts CI/CD
- Rédaction du cahier des charges avec mon coéquipier.
- Réalisation du stack de déploiement docker permettant d'avoir un environnement de développement et de production fiable et déployable facilement.
- Rédaction des utilitaires de projet (Makefile, etc...)
- Rédaction de la documentation :
 - Documentation administrateur
 - Documentation utilisateur
 - Diagramme de cas d'utilisation
 - Diagramme de contexte static
 - Modèle entité association
 - Diagrammes de séquences en tout genre pour expliquer les étapes du fonctionnement avec docker et les servers.
- Diagrammes de classe
- Documentation complète de l'environnement de développement
- Documentation sur les étapes de CI/CD
- Rédaction du guide d'installation

Architecture

Voici l'architecture des fichiers du projet :

Racine du projet

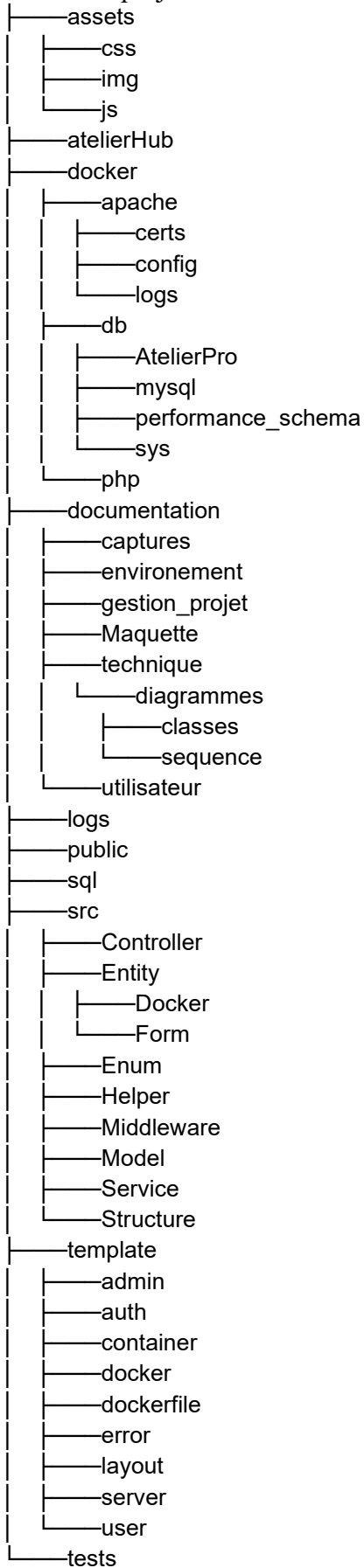
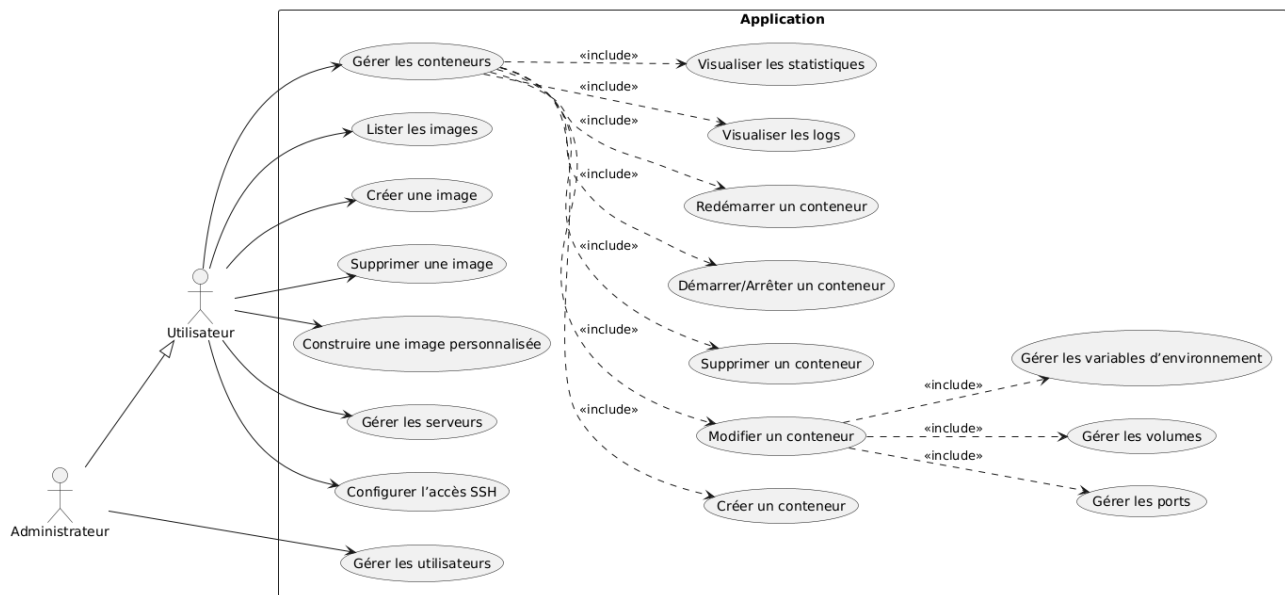
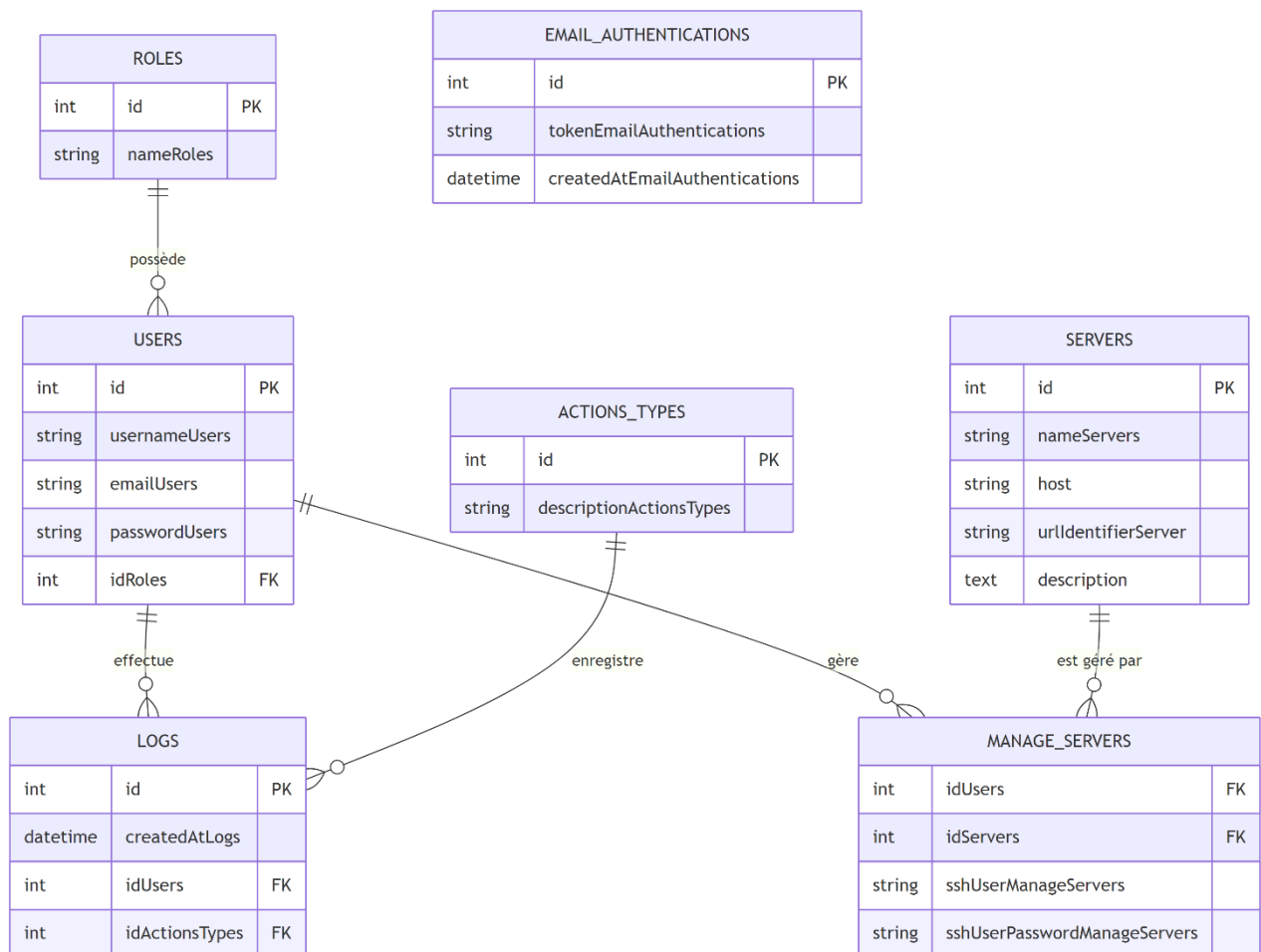


Diagramme de cas d'utilisation



Modèle entité association



Diagrammes de séquence

Diagramme de séquence des réseaux docker

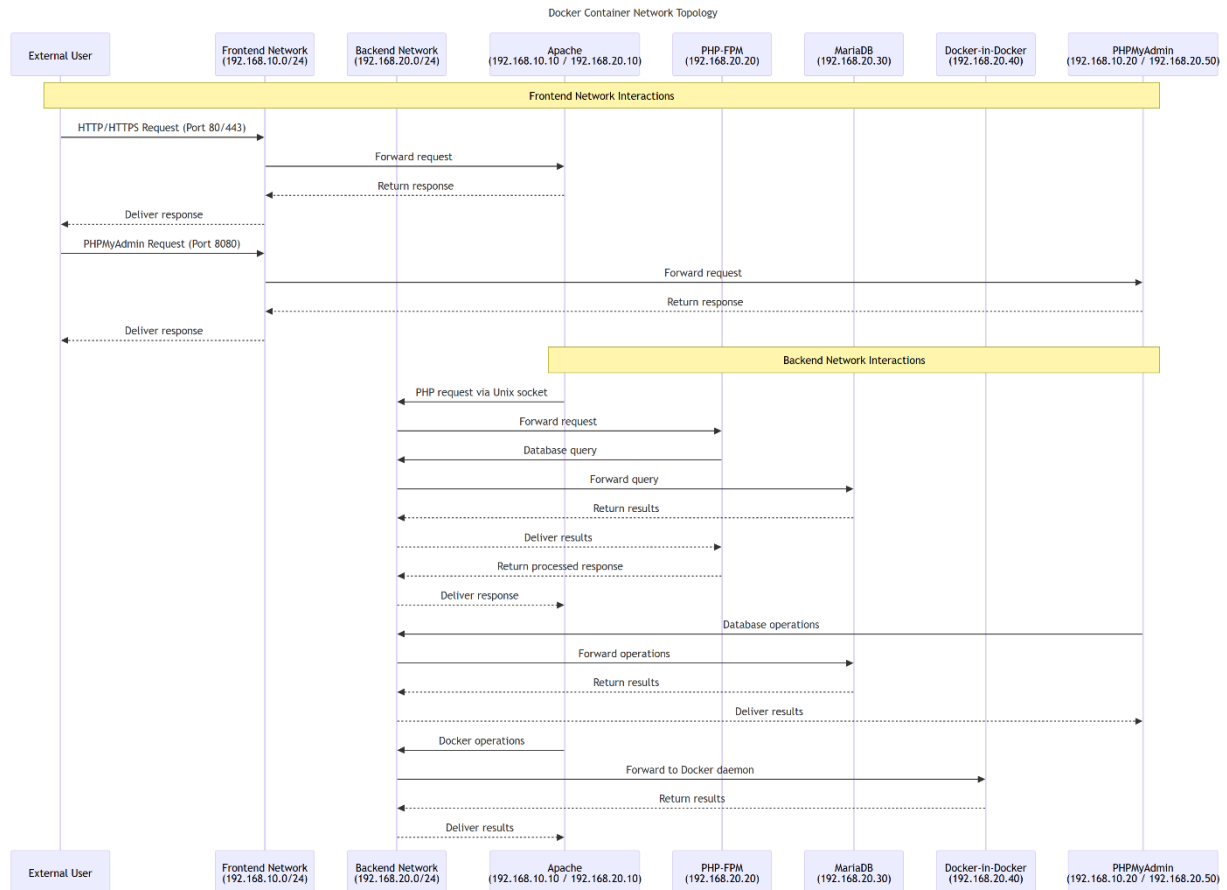


Diagramme de séquence des requêtes http

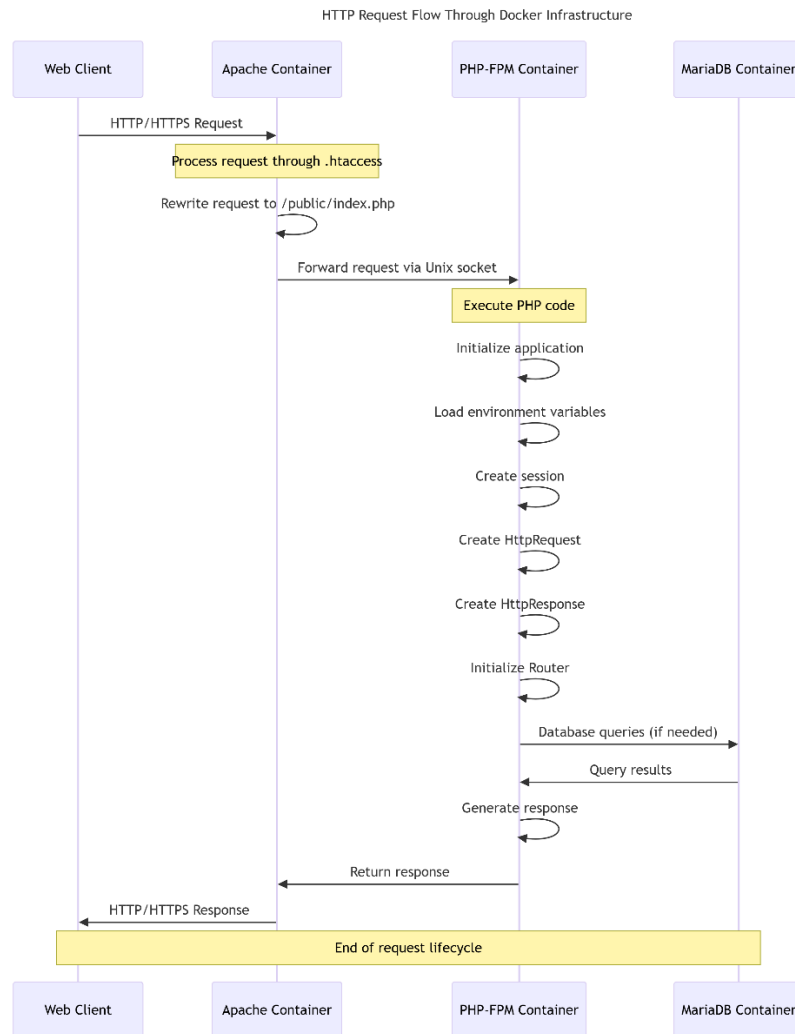


Diagramme des volumes docker

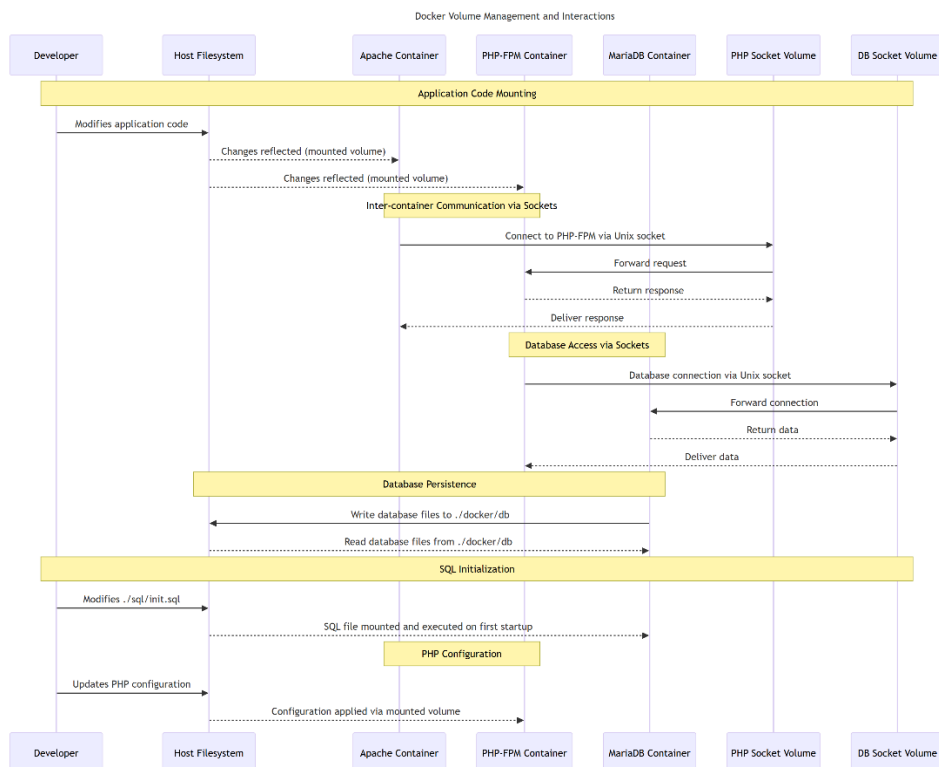


Diagramme du lancement de l'application

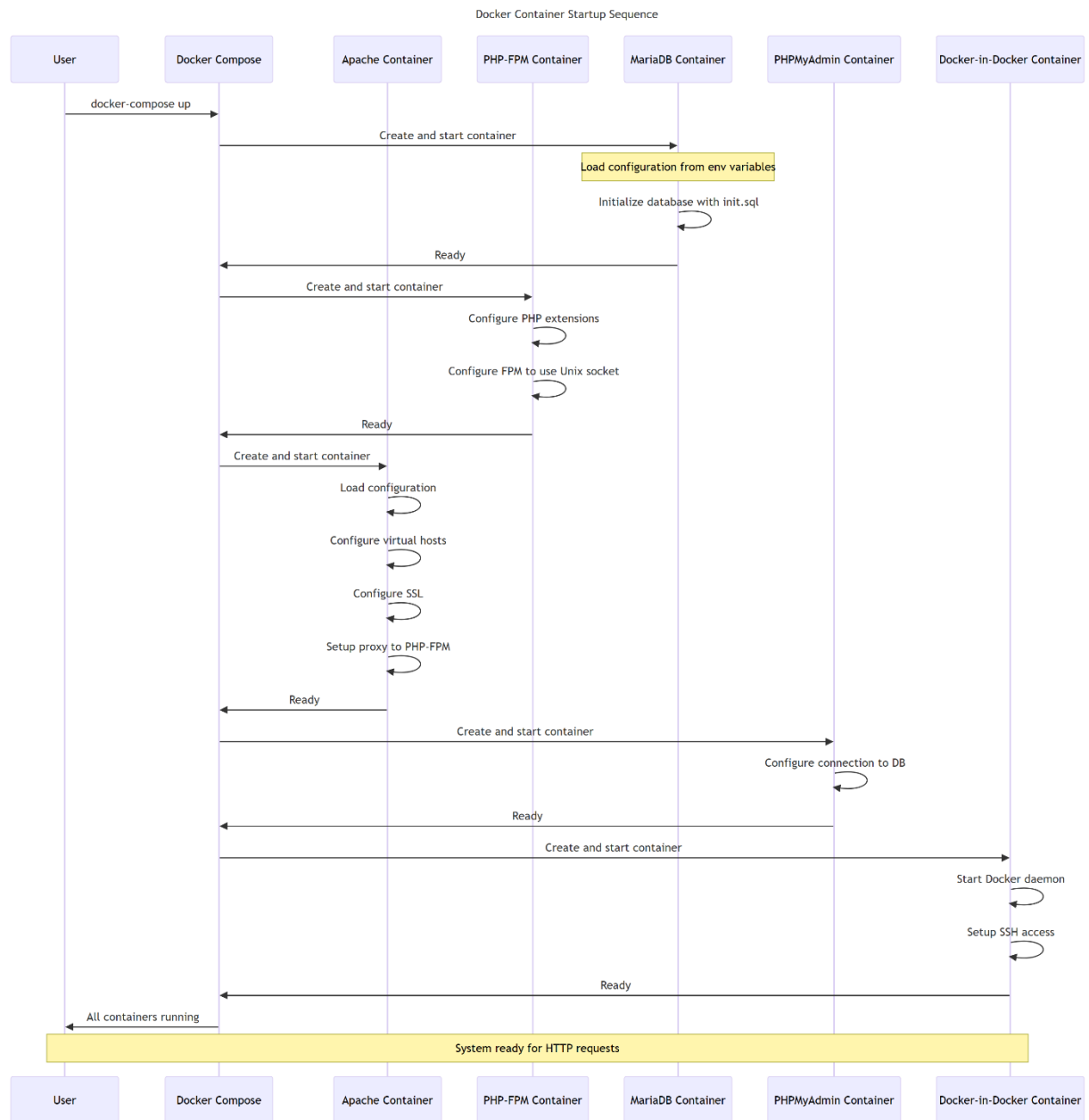
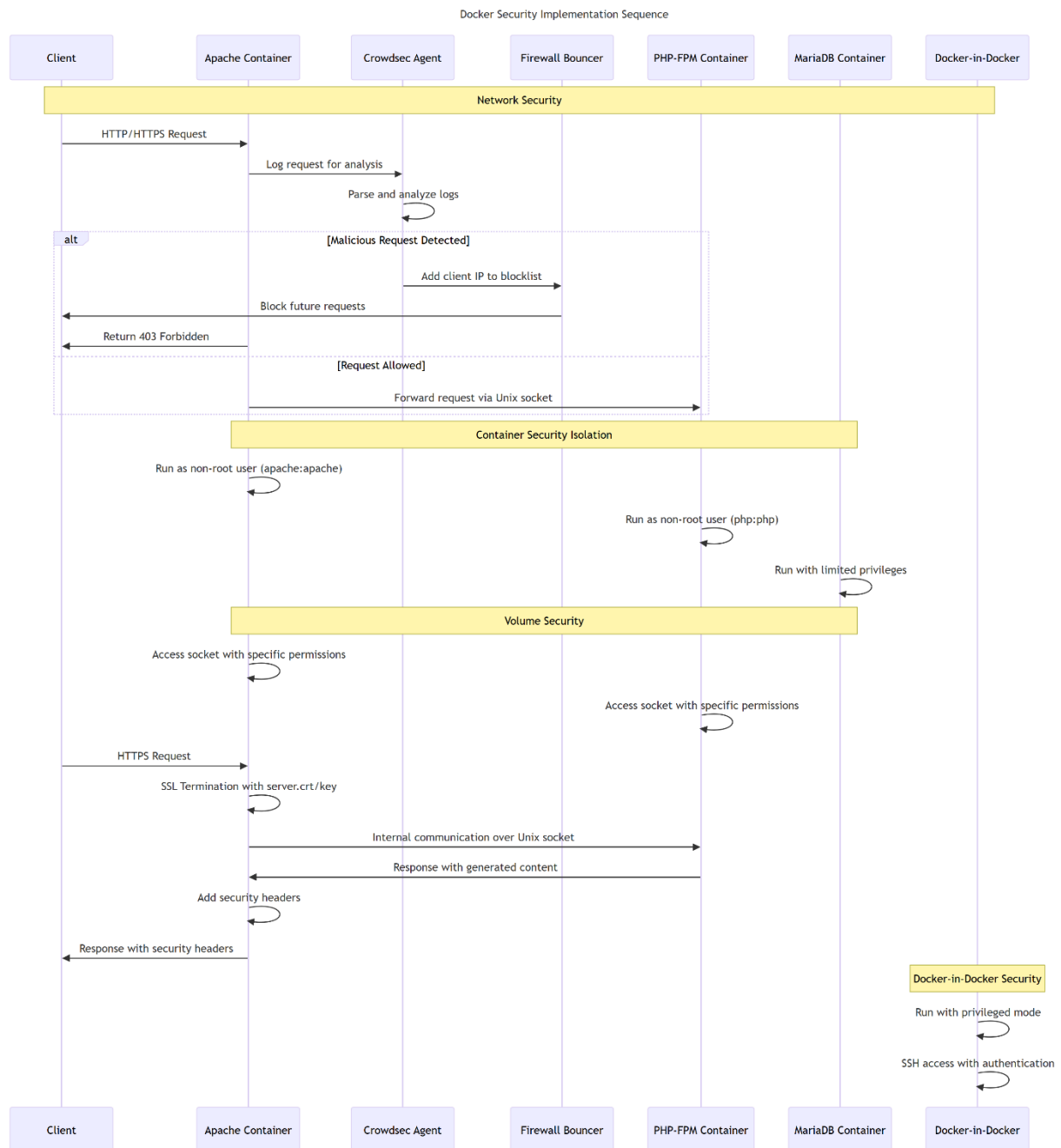
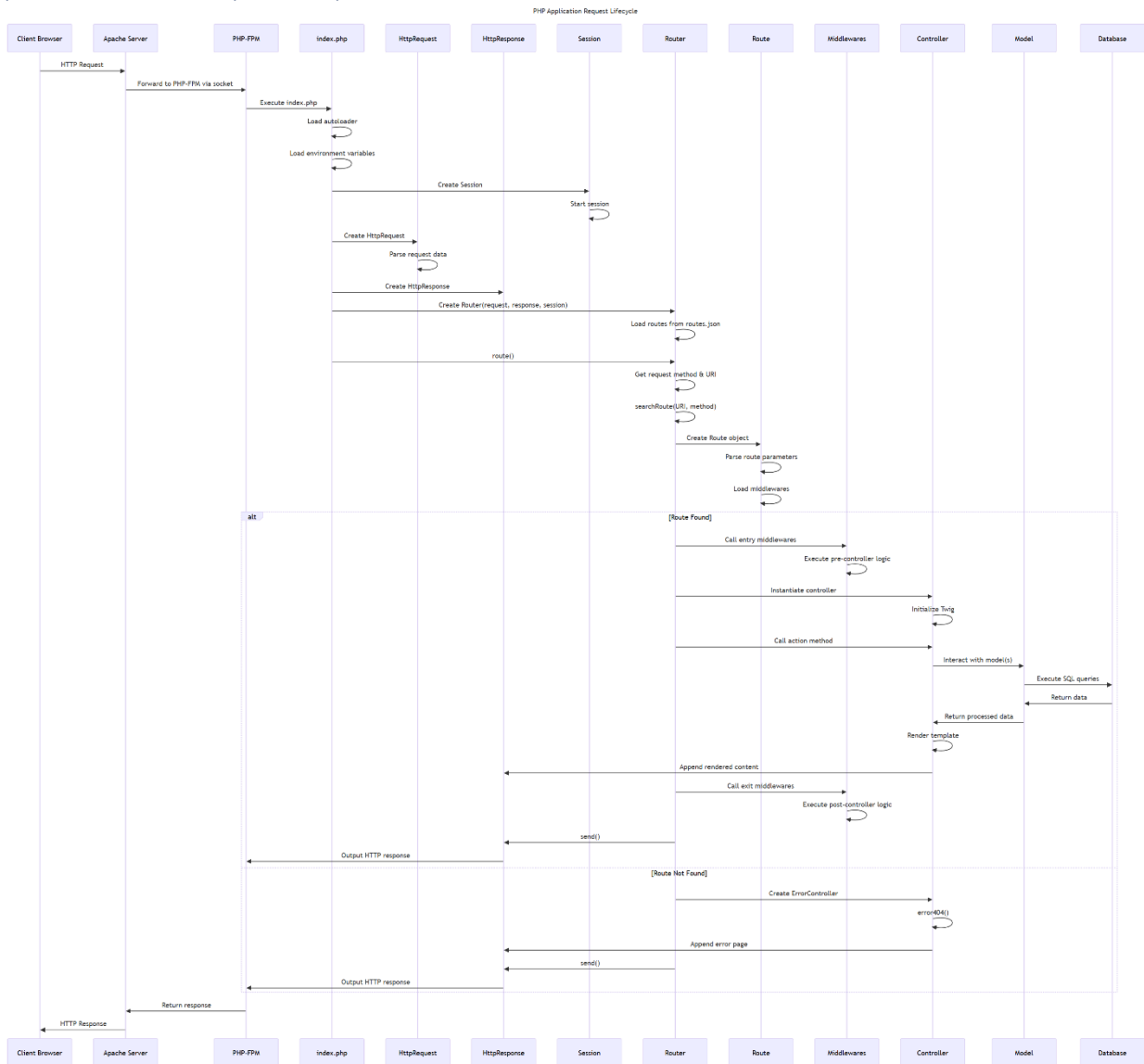


Diagramme des couches de sécurité docker



Cycle de vie de la requête http



Intégration docker dans php



Diagramme des flux de données

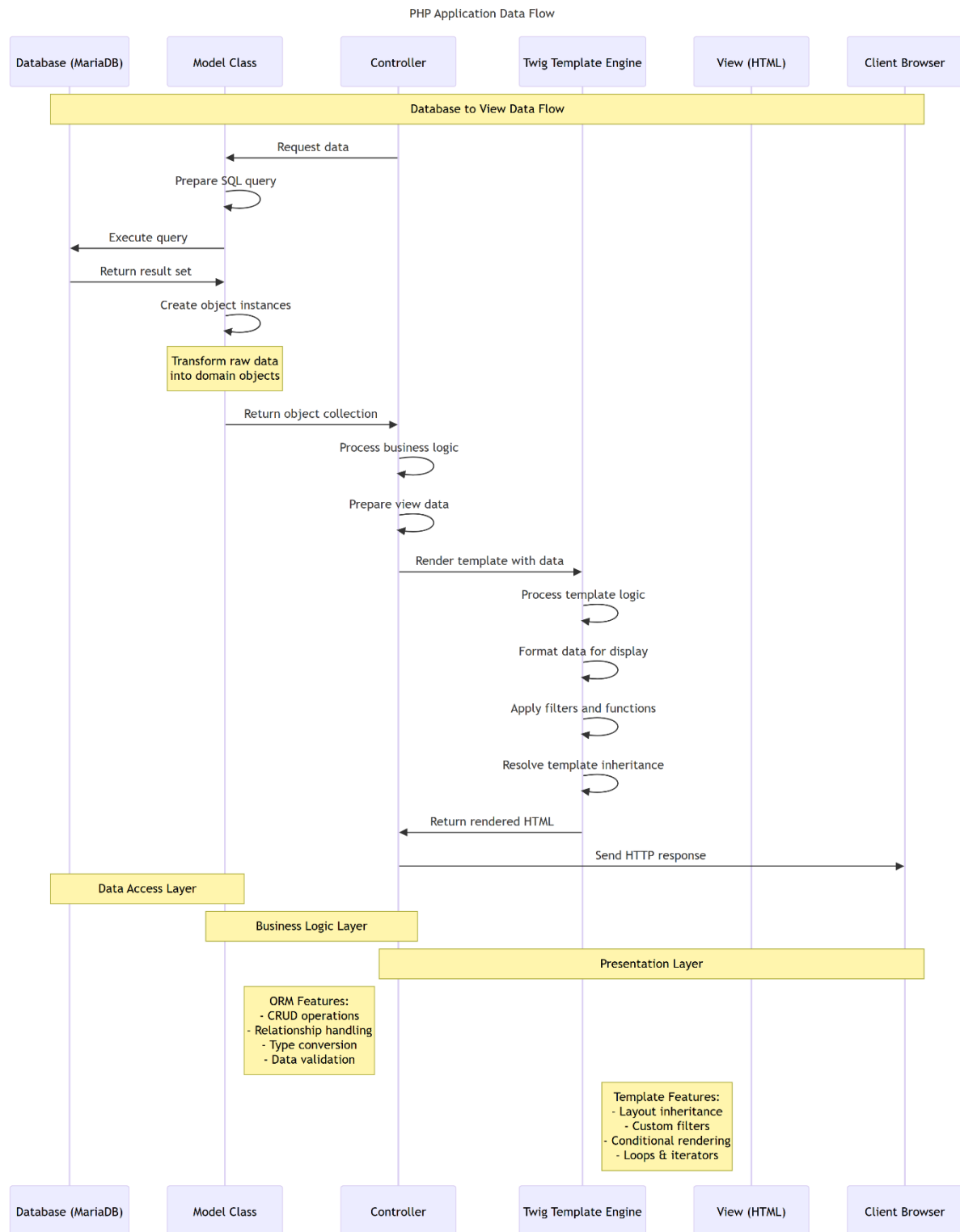


Diagramme des étapes des middlewares

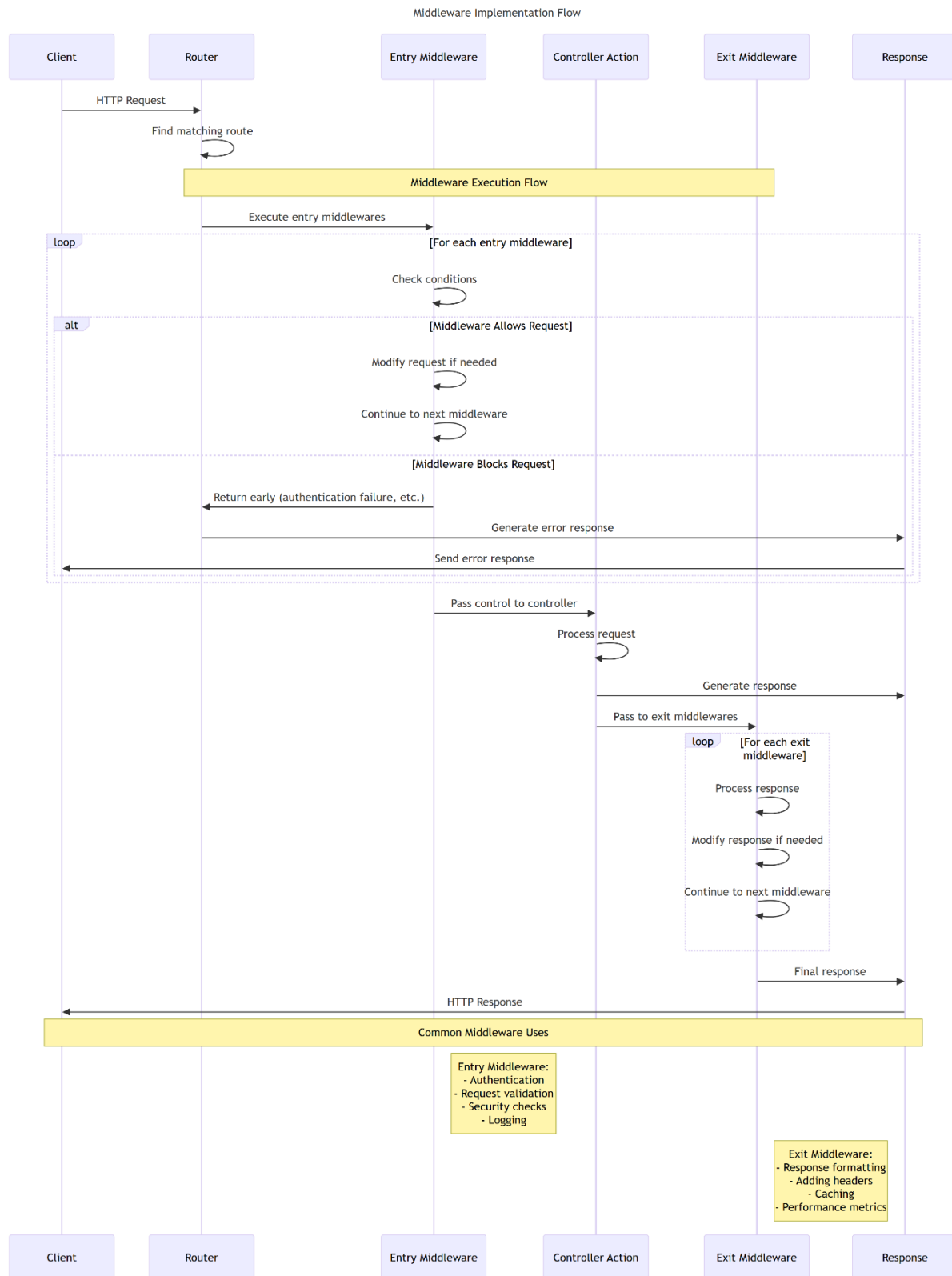
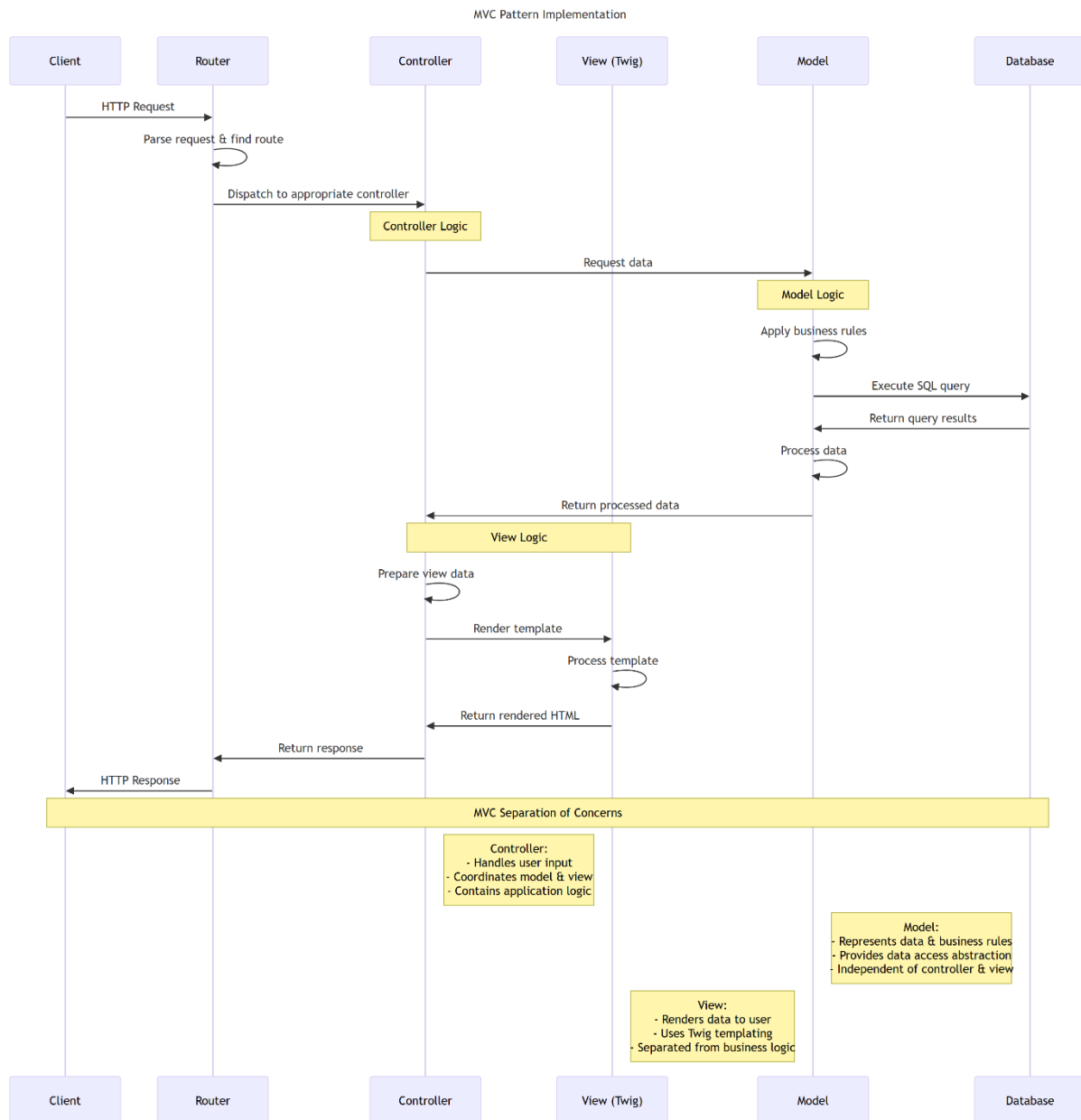


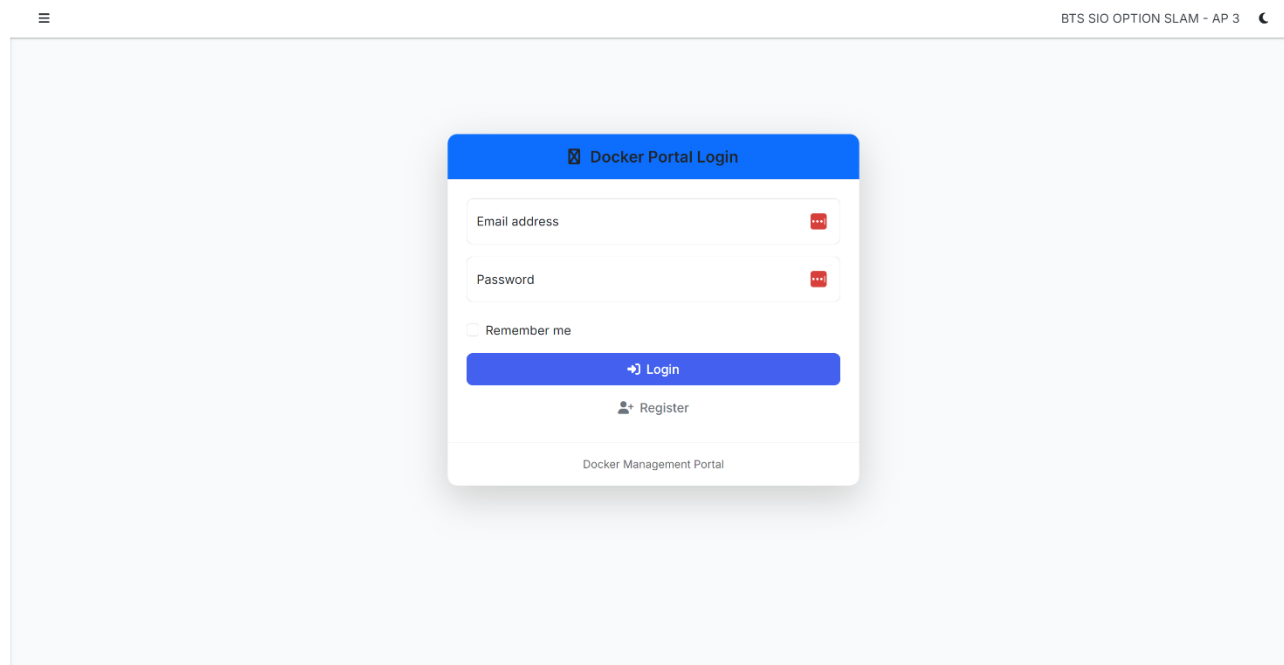
Diagramme de représentation du MVC




Captures d'écrans de l'application

Vous trouverez ci-dessous des captures d'écran de l'application :

Page de connexion



Gestion d'un conteneur

**Docker Portal**

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Retour

apache | 4503c6e87849
cci-bts-sio-23-25-atelier-professionnalisation-3-apache:latest

Ports

1. 443:443
2. 80:80

Modifier un port

Supprimer un port

Ajouter un port

Volumes


1. /var/lib/docker/volumes/cci-bts-sio-23-25-atelier-professionnalisation-3_php-socket/_data:/var/run
2. C:\Users\damie\Downloads\CCI-BTS-SIO-23-25-Atelier-Professionnalisation-3

Ajouter un volume

Modifier un volume

Supprimer un volume

Environment

**Docker Portal**

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Environment

1. PATH=/usr/local/apache2/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
2. HTTPD_PREFIX=/usr/local/apache2
3. HTTPD_VERSION=2.4.63
4. HTTPD_SHA256=88fc236ab99b2864b248de7d49a008ec2afd7551e64dce8b95f58f32f94c46ab
5. HTTPD_PATCHES=

Ajouter une variable d'environnement

Modifier une variable d'environnement

Supprimer une variable d'environnement

Commande

httpd -foreground

Modifier la commande

Stats

CPU Usage: 0.01%, Memory Usage: 12.13MiB /

Logs

Docker Portal

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Logs

```

1. 192.168.20.1 - - [24/Apr/2025:07:38:41 +0000] "GET /assets/js/editor.js HTTP/1.1" 200 8197<br />
2. 192.168.20.1 - - [24/Apr/2025:07:38:41 +0000] "GET /assets/img/favicon.png HTTP/1.1" 200 5477<br />
3. 192.168.20.1 - - [24/Apr/2025:07:38:57 +0000] "POST /dockerfile/save/nginxcustom.dockerfile HTTP/1.1" 200 15<br />
4. 192.168.20.1 - - [24/Apr/2025:07:39:01 +0000] "POST /dockerfile/save/nginxcustom.dockerfile HTTP/1.1" 200 15<br />
5. 192.168.20.1 - - [24/Apr/2025:07:39:06 +0000] "GET /dockerfiles HTTP/1.1" 200 14834<br />
6. 192.168.20.1 - - [24/Apr/2025:07:39:09 +0000] "GET /assets/img/favicon.png HTTP/1.1" 200 5477<br />
7. 192.168.20.1 - - [24/Apr/2025:07:39:12 +0000] "GET /admin HTTP/1.1" 200 11193<br />
8. 192.168.20.1 - - [24/Apr/2025:07:39:19 +0000] "GET /assets/img/favicon.png HTTP/1.1" 200 5477<br />
9. 192.168.20.1 - - [24/Apr/2025:07:39:32 +0000] "GET /logout HTTP/1.1" 302 -<br />
10. 192.168.20.1 - - [24/Apr/2025:07:39:34 +0000] "GET /login HTTP/1.1" 200 8138<br />
11. 192.168.20.1 - - [24/Apr/2025:07:39:37 +0000] "GET /assets/img/favicon.png HTTP/1.1" 200 4996<br />
12. [Thu Apr 24 08:02:26.619707 2025] [mpm_event:notice] [pid 1:tid 1] AH00492: caught SIGWINCH, shutting down gracefully<br />
13. [Thu Apr 24 08:19:47.916944 2025] [so:warn] [pid 1:tid 1] AH01574: module headers_module is already loaded, skipping<br />
14. [Thu Apr 24 08:19:47.924802 2025] [so:warn] [pid 1:tid 1] AH01574: module deflate_module is already loaded, skipping<br />
15. [Thu Apr 24 08:19:47.924848 2025] [so:warn] [pid 1:tid 1] AH01574: module proxy_module is already loaded, skipping<br />
16. [Thu Apr 24 08:19:47.924853 2025] [so:warn] [pid 1:tid 1] AH01574: module proxy_fcgi_module is already loaded, skipping<br />
17. [Thu Apr 24 08:19:47.924857 2025] [so:warn] [pid 1:tid 1] AH01574: module ssl_module is already loaded, skipping<br />
18. [Thu Apr 24 08:19:47.953839 2025] [ssl:warn] [pid 1:tid 1] AH01906: localhost:443:0 server certificate is a CA certificate (BasicConstraints: CA == TRUE !?)<br />

```

Gestion d'un serveur

Docker Portal

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Internal

Retour aux serveurs

Détails du serveur

Nom: Internal


Hôte: 192.168.20.40

Utilisateur SSH: dockeruser

Conteneurs Docker

Ajouter un conteneur

ID	Nom	Image	Statut	Ports	Actions
4503c6e87849	apache	cci-bts-sio-23-25-atelier-professionnalisation-3-apache:latest	En cours	0.0.0.0:80→80/tcp, 0.0.0.0:443->443/tcp	Détails Démarrer Supprimer
d5e25a94d3a7	phpmyadmin	phpmyadmin:latest	En cours	0.0.0.0:8080->80/tcp	Détails Démarrer Supprimer

**Docker Portal**

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Images Docker


Télécharger une image

Repository	Tag	ID	Taille	Créé
cci-bts-sio-23-25-atelier-professionnalisation-3-dind	latest	1952c9da1f41	1.56GB	2025-04-23 06:32:21 +0000 UTC
cci-bts-sio-23-25-atelier-professionnalisation-3-apache	latest	f05fb8f71803	770MB	2025-04-22 08:11:25 +0000 UTC
cci-bts-sio-23-25-atelier-professionnalisation-3-php	latest	815829976ee3	836MB	2025-04-22 08:08:25 +0000 UTC
cci-bts-sio-23-25-atelier-professionnalisation-3-composer	latest	df76e668bff8	885MB	2025-04-22 07:54:05 +0000 UTC
mariadb	latest	81e893032978	454MB	2025-02-14 06:55:09 +0000 UTC
phpmyadmin	latest	216747e3a6b7	813MB	2025-01-24 08:52:08 +0000 UTC
httpd	latest	4564ca760495	221MB	2025-01-24 00:31:17 +0000 UTC

Dockerfiles

Utiliser un Dockerfile

Gestion des serveurs

**Docker Portal**

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Gestion des serveurs

Gérez vos serveurs Docker et leurs conteneurs

+ Ajouter un serveur

Ajouter un accès

Rechercher un serveur par nom ou adresse...

Internal


192.168.20.40

Standard

Actif

→

Gestion des dockerfiles




Docker Portal


Gestion des serveurs


Gestion des dockerfiles

Gestion des utilisateurs




BTS SIO OPTION SLAM - AP 3









 Dockerfile Management

Servers / Dockerfiles


 Your Dockerfiles

+ Add Dockerfile

Name	Last Modified	Actions
 nginxcustom.dockerfile	2025-04-24 09:35:25	<div> </div>
 test.dockerfile	2025-04-24 09:35:25	<div> </div>

Q Search

Find by name...




Information

Dockerfiles are text documents that contain all commands needed to build a Docker image.
You can create, edit, and delete your Dockerfiles here, then use them to build custom images on your Docker servers.

Getting Started

1. Create a new Dockerfile using the "Add Dockerfile" button

2. Edit your Dockerfile with our powerful editor




Docker Portal


Gestion des serveurs


Gestion des dockerfiles

Gestion des utilisateurs



BTS SIO OPTION SLAM - AP 3



 Create New Dockerfile

Dockerfiles / Add New

Create Dockerfile

Dockerfile Name

e.g. web-server, python-app


.dockerfile

Enter a name for your Dockerfile. Only use letters, numbers, hyphens and underscores.


Template (Optional)

Blank


Start with an empty Dockerfile

 Nginx

Web server configuration

 PHP

PHP with Apache

 Node.js

Node.js application

Create Dockerfile

Cancel

Information

What is a Dockerfile?

A Dockerfile is a text document containing commands to assemble a Docker image automatically.
After creating your Dockerfile, you can edit it with our built-in editor to customize it for your needs.


Next Steps

1. Create your Dockerfile

2. Edit the contents with our editor

3. Use it to build a Docker image on your server

21




Docker Portal

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

 Edit Dockerfile

Dockerfiles / nginxcustom


</> Editor

SaveClose

Theme: DarkFont Size: 14pxWord Wrap

```
1 FROM nginx:alpine
2
3 # Copy custom nginx config
4 COPY nginx.conf /etc/nginx/conf.d/default.conf
5
6 # Copy static website content
7 COPY ./html /usr/share/nginx/html
8
9 # Expose port 80
10 EXPOSE 80
11
12 # Start Nginx
13 CMD ["nginx", "-g", "daemon off;"]
```

Line: 1, Column: 1DockerfileLast saved: Never




Docker Portal

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Line: 1, Column: 1DockerfileLast saved: Never

 Dockerfile Tips

Common Instructions

FROM - Base image

WORKDIR - Working directory

COPY - Copy files

RUN - Execute commands


ENV - Environment variables

EXPOSE - Port exposure


CMD - Default command

Best Practices


Templates

 Nginx


Basic Nginx web server configuration

 Node.js

Node.js application with npm


 PHP

PHP application with Apache

 Python

Python application with pip

22

Docker Portal

Gestion des serveurs

Gestion des dockerfiles

Gestion des utilisateurs

Gestion des utilisateurs

Ajouter un utilisateur

Rechercher un utilisateur...

Tous les rôles

#	Nom d'utilisateur	Email	Rôle	Actions
1	admin	admin@localhost	Administrateur	<div></div> <div></div>