

TP: Configuration d'un Pipeline CI/CD pour une Application Web PHP

1 — Compréhension et configuration de base

1.1 — Analyse du projet PHP

Il s'agit d'une application PHP simple, avec principalement `index.php`, qui va afficher deux rectangles contenant chacun du texte, et ce en appelant l'app' dédiée `ImageCreator.php`.

- `ImageCreator.php` définit la classe `ImageCreator` qui va prendre en paramètre deux couleurs et deux chaînes de caractères pour ainsi remplir l'objectif énoncé à l'instant.
- L'utilisation du gestionnaire de packages PHP « Composer » est ici nécessaire pour l'appel à la bibliothèque de gestion de date & heure « Carbon », dont l'unique but est de joindre au texte du premier rectangle, la date et l'heure à laquelle celui-ci est généré et donc en l'occurrence à peu près la date et l'heure de l'affichage de la page.

Étant donné le peu d'éléments soulignés, d'autres plus anecdotiques peuvent être évoqués :

- Une page `info.php` est présente et génère la page d'information standard de php grâce à `phpinfo()`.
- Une police d'écriture présente via le fichier `consolas.ttf` est utilisée par `ImageCreator`

Concernant la configuration et le déploiement de l'application, celle-ci est containerisée, avec Docker, via `Docker/Dockerfile`.

1.2 — Configuration de l'environnement Docker

Commençons par construire l'image

```
cd "php-devops-tp"

# Commandes docker lancées en tant qu'utilisateur root

# Construction de l'image à partir de la racine du projet
docker build --tag php-devops-tp --file Docker/Dockerfile .
# L'image a bien été créée
docker images

# Instanciation d'un container qui va tourner en arrière-plan
docker run --detach --interactive --tty \
  --publish target=80,published=127.0.0.1:9852,protocol=tcp \
  --add-host host.docker.internal:host-gateway \
  --name php-devops-tp_container php-devops-tp
# Il nous est possible d'accéder à la page via le navigateur, à l'adresse "http://localhost:9852"

# Le container est lancé
docker ps

# Possibilité d'entrer dans le container via tty avec `bash`
docker exec --interactive --tty php-devops-tp_container /bin/bash
```

Code 1. – Construction de l'image docker,
instanciation d'un container et accès au terminal du container

Nous avons à présent



Fig. 1. – Container up & running & accessible

Un problème apparaît cependant avec `index.php` comme le montre l'image ci-dessous

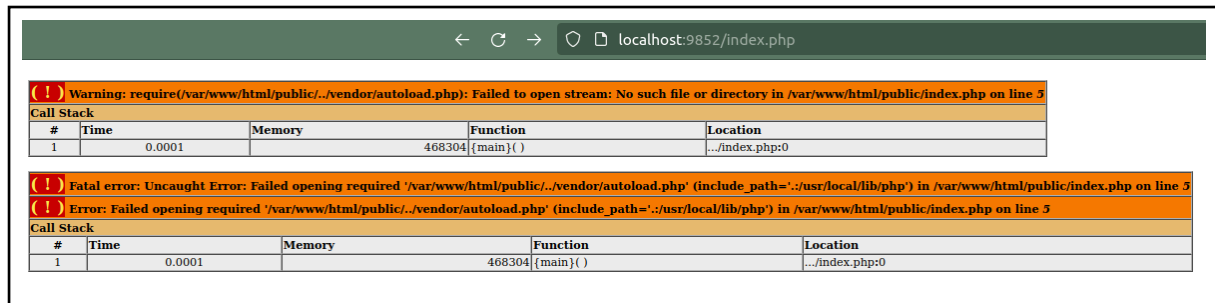


Fig. 2. – Problem with index.php

Pour le corriger, il reste à importer les dépendances avec « Composer » (`composer update`), soit manuellement avec le tty interactif une fois le container lancé, soit en ajoutant l'instruction dans le `Dockerfile`.

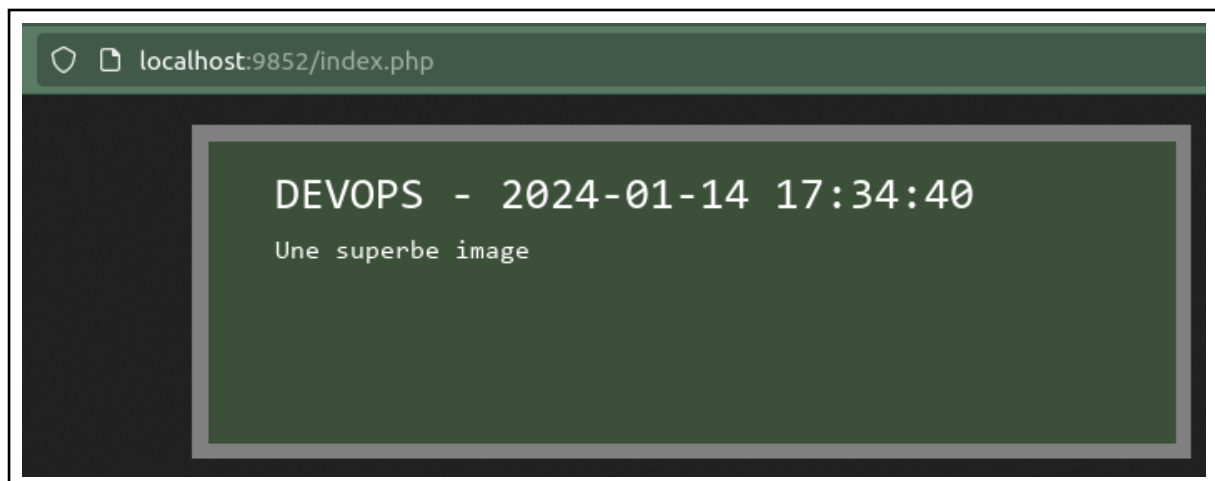


Fig. 3. – index.php fonctionnel après l'ajout des dépendances

Il sera choisi de modifier le `Dockerfile` pour que les dépendances soient déjà correctes lors de l'instanciation d'un container issu de l'image docker.

2 – Mise en place du pipeline CI/CD

2.1 – Configuration GitHub et CircleCi

Le dépôt étant à présent sur GitHub, nous allons configurer CircleCi

2.2 – Création du pipeline CI/CD

2.3 – Gestion des secrets avec Infisical

3 — Extension du Pipeline

3.1 — Ajout de jobs dévaluation de code

3.2 — Intégration de la qualité du code

3.3 — Déploiement automatisé sur AWS EC2