
1 INFORMATIONS GENERALES

Élèves :	Nom : Berchel	Prénom : Joachim
	Nom : Simoes	Prénom : Lucas
Lieu de travail :	ETML / Avenue de Valmont 28b, 1010 Lausanne	
Client	Nom:	Prénom:
Dates de réalisation :	1 ^{er} trimestre	
Temps total :	~24 périodes	

2 PROCÉDURE

- Tous les apprentis réalisent le projet sur la base d'un cahier des charges.
- Le cahier des charges est présenté, commenté et discuté en classe.
- Les apprentis sont entièrement responsables de la sécurité et sauvegarde de leurs données.
- En cas de problèmes graves, les apprentis avertissent le client au plus vite.
- Les apprentis ont la possibilité d'obtenir de l'aide externe, mais ils doivent le mentionner.
- Les informations utiles à l'évaluation de ce projet sont disponibles au chapitre 8.

3 TITRE

Docker Swarm Wordpress

4 SUJET

Déploiement de Wordpress avec Docker Swarm.

5 MATÉRIEL ET LOGICIEL À DISPOSITION

- Un PC ETML
- Accès à Internet
- Docker Desktop
- <https://hub.docker.com/>
- <https://github.com/>

6 PRÉREQUIS

Modules 347 – Utiliser un service avec des conteneurs

7 CAHIER DES CHARGES

7.1 Gestion de projet

1. Planification initial suscité de type **Kanban** (avec l'outil de votre choix)
2. **Journal de travail** (selon modèle fournit)
3. **Repo privé** sur github avec le nom : 347-docker-swarm-wordpress

7.2 Objectif du projet

Le projet consiste à déployer une instance WordPress hautement disponible et évolutive en utilisant Docker Swarm. Ce déploiement permettra de gérer le site web WordPress de manière plus efficace, avec des capacités d'orchestration, d'équilibrage de charge, de mise à l'échelle automatique, et de tolérance aux pannes.

Contexte :

WordPress est l'un des systèmes de gestion de contenu (CMS) les plus utilisés dans le monde. Toutefois, la gestion de son infrastructure peut devenir complexe lorsque plusieurs instances sont nécessaires pour répondre à des exigences de haute disponibilité, de performance, ou de résilience. Docker Swarm, en tant qu'outil d'orchestration de conteneurs, permet de simplifier et d'automatiser ces aspects.

Étapes Clés du Projet :

1. **Initialisation du Swarm :**
 - Création d'un cluster Docker Swarm avec **au moins trois** nœuds (**un manager et deux workers**) pour garantir la haute disponibilité.
 - Configuration de **2 réseau overlay** (un pour le **frontend** et un pour le **backend**) pour permettre aux services Docker de communiquer entre eux à travers différents hôtes.
2. **Déploiement des Services :**
 - Déploiement d'un service pour la base de données **MariaDB**, en utilisant un **volume persistant** pour stocker les données.
 - Déploiement du service **WordPress**, qui se connectera au service de **base de données** et servira le contenu **web**.
 - Utilisation de **secrets** Docker pour gérer les informations sensibles, telles que les **mots de passe** de la base de données.
3. **Mise en Place de l'Équilibrage de Charge :**
 - Docker Swarm offre un **équilibrage de charge** intégré qui répartit automatiquement le trafic entrant entre les différentes répliques du service WordPress.
 - Configuration d'un **service de proxy inverse** avec **Nginx** pour gérer les requêtes **HTTP** pour un accès depuis un **nom de domaine**.
4. **Mise à l'Échelle et Tolérance aux Pannes :**
 - Configuration du nombre de répliques du service WordPress pour répondre aux exigences de charge et garantir la continuité du service en cas de panne d'un ou plusieurs nœuds.

Le répertoire du code **Wordpress** sera un **volume partagé** avec **Nginx** le serveur Web. Pour ce faire, un serveur **NFS** sera nécessaire afin de pouvoir garantir la persistance des données. Les instructions pour mettre en place ce serveur **vous sera transmise par votre enseignant**.

Vous ne pouvez pas utiliser d'interface graphique pour la mise en place de la stack Docker Swarm, uniquement par CLI !

Afin de pouvoir visualiser les stacks Swarm et toutes les informations y relative, comme les nœuds, les volumes, etc. nous vous recommandons d'installer une interface open-source nommé **Swarmpit**.

Vous devez utiliser les images **officielles** de **Mariadb**, Wordpress (**php-fpm**) et **Nginx** !

Afin de faciliter la mise en place des VMs Linux, nous recommandons d'utiliser **multipass**, voir les instructions donnés par votre enseignant.

7.3 Livrables

- Un cluster Docker Swarm entièrement opérationnel.
- Un service WordPress déployé et accessible via le nom de domaine : `www.my-wordpress.ch`.
- Une release Github, à laquelle sont inclus :
 1. Les fichiers et dossiers de configurations (`docker-compose.yml`, etc.).
 2. Documentation détaillant toutes les étapes de déploiement (commandes à effectués) avec explications exhaustives dans le fichier `README.md`.
 3. Les instructions du fichier `docker-compose.yml` doivent être documentées (`#commentaires`)

Le journal de travail individuel sous Excel, nom : `jdt_nom_prénom.xls`, ainsi que le repo github seront transmis par Teams.

8 Évaluation

1. Auto-évaluation challengée par le client basé sur des éléments observables (démon, README)
2. Une évaluation individuelle (par groupe), qui comprendra une démonstration (une à ~60% du projet et une à la fin du projet).
3. Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) doit être mentionné et ne peut servir que d'inspiration à la réalisation. Chaque développeur doit être à tout moment en mesure d'expliquer le code de manière précise et convaincante. En cas d'abus, l'évaluation du projet en tiendra compte.