

# Projet d'architecture orientée services

## Rapport

Sebastien LEONCE      Alexis DA COSTA

2022-04-24

### Abstract

Rapport et présentation de notre projet d'architecture orientée services sur la création d'un SI dockerisé permettant de gérer une application de blog. Création d'une API, de microservices et utilisation d'une multitude d'outils de monitoring et d'alerte.

## 1 Equipe et participants au projet

Ce projet a été réalisé par l'équipe suivante composée de :

- Sebastien LEONCE
- Alexis DA COSTA

## 2 Présentation et objectifs du projet

Le projet consiste à créer un **SI dockerisé** permettant de gérer une **application de blog**. Ce SI est composé d'une **multitude de services permettant de gérer** l'ensemble des ressources nécessaire pour le **fonctionnement de notre application de blog** ainsi que pour **observer et monitorer** cette dernière.

Le SI est déployé sur une **multitude de conteneurs Docker** et est accessible via un port HTTPS. Pour cela, nous avons utilisé **Docker Compose**, outil permettant de définir et d'exécuter des applications Docker multi-conteneurs.

Cette application a été développée dans **une stratégie DevOps**, utilisant une **CI/CD pour la gestion des déploiements**. Une multitude de **tests unitaires** ont été réalisés afin de vérifier le bon fonctionnement de notre application, de nos APIs ainsi que microservices.

## 3 Retours sur le projet

Au debut de notre projet, nous avons établi avec notre professeur, un plan de travail et une liste d'éléments attendus.

Voici ci-dessous la liste de ces éléments :

- **NECESSAIRE** : Développer une API permettant de gérer l'ensemble des ressources de notre application de blog, avec des routes permettant de gérer les différentes ressources liées à la gestion des articles, des commentaires, des utilisateurs et du mécanisme d'authentification. **FAIT**
- **NECESSAIRE** : Développer un microservice permettant de gérer le mécanisme d'alerte et notification par mail aux utilisateurs de notre application de blog, exploitant des filesMQ. **FAIT**
- **NECESSAIRE** : Définir un CI/CD. **FAIT**
- **NECESSAIRE** : Définir des tests unitaires et/ou une collection de tests Postman testant le bon fonctionnement de notre API et microservice. **FAIT**
- **OPTIONNEL** : Utiliser un système de monitoring pour observer et monitorer l'application, par prometheus et/ou grafana. **FAIT**
- **NECESSAIRE** : Utiliser un système de conteneurisation pour nos services et nos microservices, par docker ou docker-compose. **FAIT**
- **NECESSAIRE** : Utiliser un système de gestion de version pour nos microservices, par gitlab ou github. **FAIT**
- **OPTIONNEL** : Utiliser un système d'orchestration pour nos microservices, par kubernetes ou mesos. **NON-FAIT, par manque de temps et d'expérience sur le sujet. Tentative faite d'utiliser Docker-Swarm ou encore l'outil Kompose infructueuse et non concluante. Nous avons choisi de nous concentrer sur la bonne mise en place de nos services utilisant docker-compose.**
- **OPTIONNEL** : Développer un frontend de l'application de blog, permettant de gérer les articles, les commentaires, les utilisateurs et le mécanisme d'authentification, utilisant notre API. **NON-FAIT, par manque de temps. Le front sera réalisé dans le cadre du cours de Web Avancé**
- **OPTIONNEL** : Utiliser un provider PasS Cloud pour déployer en production notre application de blog. **FAIT**

## 4 Retours sur le cours

Nous avons fortement apprécié le cours d'Architecture Orientée Service. Cela nous a permis d'approfondir des notions de base sur le développement d'un SI, de l'architecture et de la gestion de projet, de l'utilisation de Docker et de Docker-Compose, dans une architecture DevOps, travailler sur des outils de monitoring et d'alerte, d'utiliser des outils de CI/CD et de gestion de version, d'utiliser des outils de conteneurisation et d'orchestration, d'utiliser des outils de déploiement en production. Ces notions sont très utilisées au sein d'une grande partie de nos entreprises.

Un TP sur Kubernetes ou Docker-Swarm peut être très intéressant pour les futures années.

**NB :** Pour accéder aux services d'administration de notre SI en localhost tel que le dashboard de monitoring...

- Identifiant de connexion : **admin**
- Mot de passe : **admin**