

Proposition de Stage Master II ou projet de fin d'études 2018

Sujet Stage

Architecture microservices pour le développement d'applications cloud natives. Application au développement d'une application de réservation en ligne.

Informations complémentaires

Encadrant(s) : Jamal Malki (*), Jean-Philippe Baron (**)

Cadre de coopération : L3i (*) – aYaline (**)

Date de début du stage : à partir de janvier 2018

Durée du contrat : 5 mois (minimum)

Stage rémunéré

Contexte

Le stage se déroule dans le cadre du projet PLAIBDE : « Plateforme Intégrée Big-data pour les Données Entreprise ». Ce projet fait partie du programme FEDER-FSE 2014-2020 porté par la région Nouvelle Aquitaine.

Ce projet est dirigé par le consortium suivant :

1. L'entreprise aYaline¹ : partenaire industriel et chef de fil du projet
2. Le laboratoire L3i² : partenaire scientifique, le L3i (laboratoire Informatique, Image, Interaction) fait partie de l'université de La Rochelle
3. Le laboratoire LIAS³ : partenaire scientifique, le LIAS (Laboratoire d'Informatique et d'Automatique pour les Systèmes), fait partie de l'ENSMA (École Nationale Supérieure de Mécanique de d'Aéronautique – Futuroscope), université de Poitiers

L'objectif du projet FEDER-PLAIBDE est le développement d'un écosystème Big-Data métier dans les domaines d'activités relevant de l'expertise de l'entreprise aYaline : E-Commerce, E-Tourisme, E-Collectivité et d'autres.

¹ <http://www.ayaline.com>

² <http://l3i.univ-larochelle.fr>

³ <https://www.lias-lab.fr>

Objectifs du stage

L'avènement des architectures applicatives basées sur des microservices est considéré comme une avancée remarquable dans le développement et le déploiement d'applications. Les microservices sont un élément essentiel d'un certain nombre de progrès importants qui changent la nature de notre façon de travailler. Des techniques de développement logiciel agiles, le cloud computing, la culture DevOps, l'intégration continue et le déploiement continu, l'utilisation conjointe des microservices et des conteneurs considérée comme le fondement de l'architecture CaaS (Container as a Service) à la base de toute application cloud native : toutes ces avancées sont utilisées au côté des microservices pour révolutionner le développement et la livraison des applications.

Suivant une architecture microservices, une application est développée, ou repensée, sous la forme d'une collection de mini-services distincts qui communiquent d'une manière bien définie, via des API HTTP par exemple. Chaque microservice est une unité fonctionnelle précise qui implémente des fonctionnalités métiers spécifiques. Chaque microservice est autonome, maintient son propre espace de données (ce qui a des implications importantes), et peut être mis à jour indépendamment des autres. Passer à une approche basée sur les microservices rend le développement d'applications plus rapide et plus facile à gérer. Les changements peuvent être effectués et déployés plus rapidement et plus facilement. Une application conçue comme une collection de microservices est plus facile à exécuter sur plusieurs serveurs avec équilibrage de charge, ce qui facilite la gestion des demandes des ressources, tout en réduisant les temps d'arrêt causés par des problèmes matériels ou logiciels.

Dans une architecture microservices containérisée, on peut déployer, modifier, monter en capacités et redémarrer chaque microservice de façon indépendante, généralement à partir d'un système automatisé permettant des mises à jour fréquentes en production, sans aucun impact sur le client. Grâce à la virtualisation au niveau du système d'exploitation (OS), on peut diviser une instance d'OS de façon dynamique en plusieurs conteneurs isolés, chacun possédant son propre système de fichiers en écriture et quota de ressources. Le faible impact de la création et suppression des conteneurs, font d'eux un vecteur idéal pour le déploiement de microservices individuels.

L'objectif de ce travail est la mise en place d'une architecture et d'un environnement de développement et de déploiement de la nouvelle génération des applications métiers de l'entreprise. Fondamentalement, ce travail doit reposer sur le constat que l'architecture microservices est un ensemble de concepts et de pratiques indépendantes de toute technologie. Par conséquent, aucune définition ou spécification standard claire n'est disponible. Dans ce contexte, on doit apporter des solutions à des problématiques importantes comme :

- Définir l'API Gateway : le point d'entrée unique pour l'ensemble de toute application basée sur les microservices, présentant l'API de chaque microservice ;
- Définir la communication inter-processus dans l'architecture microservices (IPC) : parmi toutes les solutions préconisées, on s'intéresse particulièrement aux IPC utilisant REST ;
- Proposer une approche pour la découverte de services dans une architecture de microservices : lorsque les services fonctionnent dans un environnement dynamique, les identifier lorsqu'on en a besoin n'est pas un problème trivial ;
- Gérer les données : chaque microservice possède son propre magasin de données

physique, le principe de la persistance polyglotte permettant à plusieurs moteurs de base de données de s'exécuter sous chacun des microservices. Par conséquent, il est possible de choisir tel ou tel magasin de données pour tel ou tel service, au lieu de stocker toutes les données dans un système de gestion de base de données relationnelle (SGBDR) d'entreprise. Ce principe introduit certes de la flexibilité, mais augmente la complexité.

Le travail sur l'architecture microservices à mettre en place est mené en parallèle avec le travail de développement et déploiement d'une application multimodales de réservation de produits touristiques en ligne. Les étapes de validation de l'architecture dans l'environnement de développement et de déploiement sont le pivot central de ce travail.

Prérequis

Ce sujet d'adresse aux étudiant(e)s en Master 2 Informatique, ou élève de dernière année d'une école d'ingénieur en informatique.

Vous êtes rigoureux dans votre travail mais aussi créatif avec une forte envie d'apprendre et de vous investir dans un projet de développement d'applications microservices avec différentes technologies : Symfony, Angular avec TypeScript, SQL, Scala et autres

Les compétences et les connaissances dans le domaine de gestion des conteneurs seront appréciées.

Candidature

Merci d'adresser votre dossier de candidature à : jmalki@univ-lr.fr, jpbaron@ayaline.com

Le dossier de candidature doit contenir :

- ✓ le CV
- ✓ la lettre de motivation
- ✓ tout autres documents pouvant appuyer la candidature