Linkedin : vade retro Cloud public, vive Docker

Linkedin a commencé en 2003 avec une simple application Java et un serveur web. Mais le réseau social professionnel a bien changé en accueillant aujourd'hui des millions de personnes. Et en suscitant l'intérêt de Microsoft, qui l'a racheté en juin pour 26,2 milliards de dollars. Nos confrères de Networkworld ont discuté avec le directeur de l'ingéniérie chez Linkedin, Steve Ihde, pour évoquer l'évolution de l'infrastructure du réseau social.

« *La croissance du site a été une aventure* », constate le responsable. Et d'ajouter : « *au fil des ans, il y a eu des points d'inflexion dans l'ingénierie de l'infrastructure et le développement des applications* ». Ainsi, en 2011, le site commence à ressentir les premiers problèmes d'une application monolithique devenue trop complexe à gérer. Les équipes ont du mal à coordonner les montées de version toutes les deux semaines.

Une première vague de micro-services

Après réflexion, le projet Inversion est lancé avec comme credo la décentralisation des services composant l'application et une réflexion sur la façon de développer du nouveau code. « *Nous avons essentiellement révisé le processus de publication* », précise Steve Ihde. Avec ce projet, Linkedin a adopté une approche basée sur les micro-services. Chacun des milliers de services composant l'application a été géré de manière indépendante avec un responsable et une équipe qui publient des nouvelles fonctionnalités quand elles sont prêtes.

Une stratégie qui n'est pas facile à gérer, reconnaît le responsable, car de nombreux services dépendent les uns des autres. Donc, à chaque mise à jour d'un des éléments, les autres doivent suivre. « *Parfois des dizaines de mises à jour doivent être faites simultanément et cela devient difficile de mettre en musique tout cela* ». Pour franchir ce cap de l'accumulation des micro-services, Linkedin est passé par un autre point « d'inflexion » : l'adoption des conteneurs.

Un Cloud privé pour un PaaS maison

« *Nous avons réalisé que nous n'utilisions pas efficacement les ressources matérielles. Nous étions dans un système de gestion manuelle d'allocation des ressources. Une méthode qui fonctionne, mais qui n'est pas optimisée* », constate Steve Ihde. En 2014, le réseau social a lancé son Cloud privé, LinkedIn Platform as a Service (LPS), utilisé pour la gestion de développement de nouvelles applications et l'automatisation de l'allocation de ressources matérielles. Au sein de LPS, il existe deux éléments dont Rain, une plateforme d'automatisation de l'infrastructure via des API. Concrètement, les développeurs écrivent leur code et l'API se charge d'allouer la quantité de mémoire et de CPU nécessaire.

Le second composant s'appelle Maestro, défini par Steve Ihde comme « *le conducteur* » de LPS. Il s'agit d'une plateforme d'automatisation au-dessus de Rain. Elle gère les processus d'ajout de

nouveaux services dans l'application, la configuration avec d'autres services, le routage du trafic au bon endroit, l'intégration des fonctionnalités avec l'ensemble du système.

Vade Retro Cloud public

Rain et Maestro ont été conçus pour gérer le code développé et encapsulé dans des conteneurs, motorisés par Docker. Linkedin a choisi de faire fonctionner LPS sur un serveur nu (bare metal), dans un environnement non virtualisé avec des conteneurs. Donc pas d'hyperviseurs, pas d'OS multiples et pas de machines virtuelles. Cette approche à travers les conteneurs donne à l'équipe d'ingénierie une plus grande granularité notamment pour les contrôles de sécurité. Si un incident (bug ou piratage) apparaît sur le composant, les effets resteront circonscrits à celui-ci. Pour cela, il faut configurer les domaines des conteneurs dans le kernel Linux, le processus des conteneurs peut alors être masqué aux autres conteneurs du même hôte. Chaque conteneur dispose de son propre namespace réseau et de son adresse IP.

Une politique qui pourrait se décliner sur un Cloud public, évitant ainsi à Linkedin de gérer ses infrastructures. Qui plus est, depuis le rachat par Microsoft, le nouveau propriétaire pourrait être tenté de basculer ces plateformes sur son Cloud Azure. Si Steve Ihde ne souhaite pas commenter les conséquences du rachat, il précise qu'avec son équipe, ils ont exploré déjà cette piste du Cloud public. « *Ce n'est tout simplement pas rentable à notre dimension* », dit-il. Et d'ajouter : « *nous pouvons répondre nous-mêmes à nos besoins* ». Une fin de non-recevoir pour les équipes Cloud de Microsoft ?

A lire aussi :

Linkedin s'attaque aux « méchants » robots du web
LinkedIn s'affiche au plus haut, avant de basculer chez Microsoft

crédit photo @vallepu-shutterstock