BlaBlaCar généralise les conteneurs et préfère Rocket à Docker

Une production informatisée généralisée sur les conteneurs et le choix de Rocket plutôt que celui de Docker. Cette seule phrase suffit à résumer le caractère novateur du projet mené par BlaBlaCar sur toute la seconde partie de 2015. « *Quand on a lancé la société, aucune offre de Cloud sérieuse n'était disponible*, explique Simon Lallemand, ingénieur système au sein de BlaBlaCar. *On a donc pris l'habitude de gérer nous-mêmes le matériel informatique et le réseau.* » Une culture que la start-up entend bien conserver, même si, voici environ 18 mois, la société s'est lancée dans une rationalisation de ses infrastructures, avec aujourd'hui seulement « *deux modèles de serveurs achetés par racks entiers* ».

Restait à y répartir les différents services opérés afin d'optimiser la charge de ces machines. Même si elles goûtent à VMware et KVM, les équipes de production de la start-up décident de sauter l'étape de la virtualisation : « *Plutôt que de construire un Cloud privé, on a fait le choix de partir directement sur les conteneurs qui sont davantage centrés sur les applications qu'on fait tourner et qui ont un impact moindre sur les performances de la machine* », résume Simon Lallemand.

Plus de 3 000 conteneurs

La réflexion sur le sujet est entamée dès le début de l'année 2015. Une petite équipe de 4 à 5 personnes s'empare du dossier au printemps de la même année. Contrairement à d'autres organisations, le projet vient bien ici des ingénieurs système, et non des développeurs. « *On savait que, de toute façon, on n'y couperait pas. Alors on a préféré prendre les devants car nous avions vu ailleurs quelques expériences malheureuses avec des codes pas très stables arrivant en production* », explique l'ingénieur.

Et BlaBlaCar ne chôme pas : dès septembre-octobre 2015, les premières applications sur conteneurs passent en production. « *On a commencé par le plus difficile – les bases de données, les composants stateful – en se disant que si nous y parvenions, le reste serait plus simple* », résume Simon Lallemand. Fin novembre 2015, 80 % des services de BlaBlaCar, dont tous les services critiques, fonctionnent au sein de conteneurs. Depuis, les équipes de la start-up se sont attaquées aux applications restantes, des services secondaires pour l'activité du spécialiste du covoiturage. Au total, entre 3 000 et 4 000 conteneurs sont aujourd'hui en production.

La 'prod' de BlaBlaCar vote Rocket

Plus surprenant encore, BlaBlaCar n'a pas fait le choix de la solution Docker aujourd'hui dominante dans le monde des conteneurs. « *Nous avons mené nos premiers tests sur Docker car, effectivement, c'est cette technologie qui était soutenue par la communauté la plus importante* », se remémore Simon Lallemand. Une déception : les équipes d'exploitation se trouvent confrontées alors à des problèmes de stabilité et à des difficultés dans la gestion du réseau. « *A l'époque, côté réseau, Docker était un enfer* », résume Nicolas Blanc, le responsable de l'équipe architecture de BlaBlaCar, croisé lors de TIAD (The Incredible Automation Day), un événement organisé par la société de service D2SI

où la start-up spécialiste de covoiturage présentait son projet.

Témoins de l'émergence de Rocket (renommé rkt mi-2015), la solution alternative conçue par CoreOS, les équipes de BlaBlaCar s'intéressent dès lors à ce projet. Très tôt. Les premiers tests sont menés sur la version… 0.2. « *Rocket s'attaquait aux limitations que nous avions identifiées avec Docker. En particulier, la solution éliminait le recours à un daemon (processus s'exécutant en arrière-plan, NDLR) et abordait le volet réseau de façon très modulaire*, raconte Simon Lallemand. *Et dès les premières versions que nous avons testées – pourtant des moutures très alpha -, nous avons pu travailler sur une technologie déjà stable*. » BlaBlaCar trouve également face à lui des équipes de CoreOS très attentives à ses attentes. Logique : la start-up française serait à ce jour le plus gros déploiement de la technologie rkt dans le monde. « *En définitive, la plus grosse difficulté que nous avons rencontrée concernait les changements des API de Rocket, qu'il nous fallait répercuter, et pas du tout la stabilité du runtime* », reprend l'ingénieur. Convaincues, les équipes d'exploitation de BlaBlaCar passent Rocket en production dès la version… 0.7.

Bientôt l'allocation automatique

D'abord testée sur les nouveaux serveurs que déploie la start-up – sur lesquels est installée la distribution Linux CoreOS, optimisée pour les conteneurs -, la solution est aujourd'hui généralisée. Pour Simon Lallemand, ce passage aux conteneurs génère de gros gains de productivité : « *Dans notre équipe, seulement deux ou trois personnes se soucient encore du matériel, plus les autres. Les gens sont devenus plus autonomes.* » Ajouter un service, comme un serveur Web frontal pour encaisser un pic de charge, se traduit par une modification d'un fichier de configuration. « *Pour créer de nouveaux services, on est passé d'un ordre de grandeur de la semaine à une durée de l'ordre de l'heure ou de quelques minutes* », résume l'ingénieur. Prochaine étape : mettre en place un système automatique d'allocation des ressources, via l'orchestrateur Kubernetes, dont le déploiement est attendu fin 2016 ou début 2017. Depuis sa version 1.3, Kubernetes, issu de Google, supporte Rocket. Pour l'instant, les équipes de BlaBlaCar se contentent de deux outils maison, Dgr et Ggn, dédiés respectivement à la construction et au déploiement de conteneurs. Tous deux sont disponibles sur Github (ici et là).

S'il fluidifie le travail des équipes d'exploitation, le passage aux conteneurs implique des changements dans l'outillage. Premier chantier : celui de la communication entre services. « *C'est un des points les plus délicats*, avertit Simon Lallemand. *Car, avec les conteneurs, la topologie des services change beaucoup plus vite qu'avec une infrastructure classique.* » Impossible donc de se contenter des mécanismes qu'utilisaient jusqu'alors BlaBlaCar, des mécanismes basés sur Chef. La start-up a donc mis en œuvre un processus de découverte de services (Service Discovery), inspiré de l'outillage développé par AirBnB et réécrit en langage Go (Go-Synapse, le routeur, et Go-Nerve, pour le suivi des statuts). « *C'est tout un nouveau mécanisme qu'on a dû mettre en place, sans modification de l'application BlaBlaCar* », précise l'ingénieur système.

Fournir les conteneurs aux développeurs

Autre chantier : le monitoring. Si la start-up a, dans un premier temps, conservé ses outils traditionnels, elle a commencé à déployer, courant 2016, Prometheus, un outil Open Source

capable d'agréger toutes les métriques d'un même service. « *Avec le passage aux conteneurs, on a besoin d'un monitoring orienté vers les services et non plus vers les hôtes* », plaide Simon Lallemand.

Si, chez BlaBlaCar, la révolution des conteneurs vient de la production – ce n'est pas si courant -, elle n'ignore pas pour autant les développeurs. Ces derniers bénéficient déjà, sur leurs postes de travail, des modèles de conteneurs exploités en production. « *Nous allons leur fournir de plus en plus d'outils pour qu'ils puissent créer directement leurs services en conteneurs* », résume Simon Lallemand.

A lire aussi :

Pourquoi BlaBlaCar migre ses bases de données sous MariaDB

Comment BlaBlaCar a automatisé sa production IT

BlaBlacar est plébiscité par ses salariés, selon Glassdoor