Cloud Foundry : la tête au DevOps et les yeux sur Kubernetes

Ajout de code, installation des *middlewares*, conteneurisation, création du point de terminaison SSL et des systèmes de journalisation… La commande cf push est puissante. Mais a-t-on systématiquement besoin de toutes ses fonctionnalités ?

La Cloud Foundry Foundation a constaté que non. Et a adapté l'approche en conséquence, avec la dernière version (v7) de l'interface en ligne de commande associée à son PaaS.

Passée récemment en phase de disponibilité générale, elle repose sur l'API Cloud Foundry v3 – sauf pour les *plug-in*, qui resteront sur la v2 jusqu'à la prochaine version majeure du CLI.

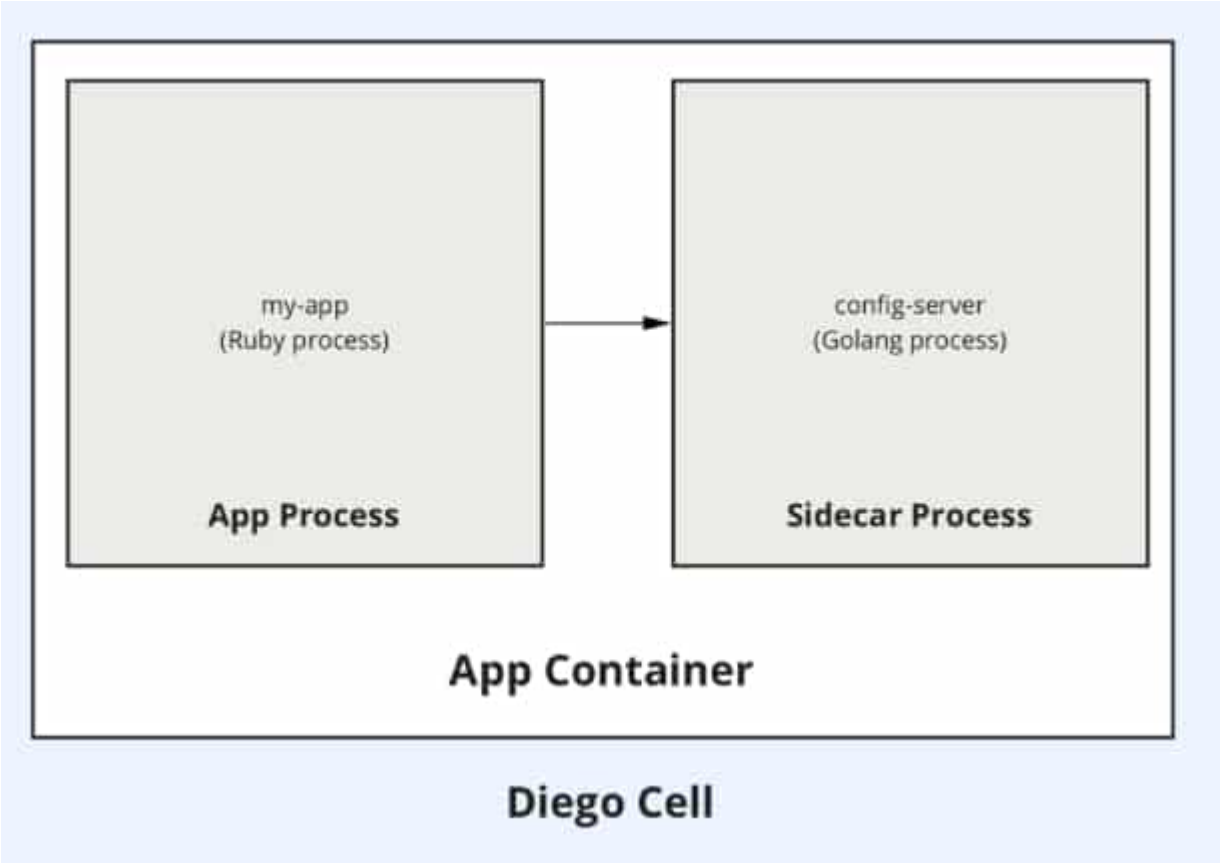
L'ensemble offre un contrôle plus fin sur cf push, en permettant de n'exécuter que certaines sous-commandes. Par exemple, mettre à jour un système tiers, envoyer des données d'audit à un service externe ou analyser un *droplet* (artefact exécutable) avant déploiement.

Autre élément introduit avec le CLI v7 (mais utilisable sur le CLI v6 en exploitant l'API v3) : le déploiement en continu. Cette technique destinée à minimiser les indisponibilités d'applications n'est pas une nouveauté sur Cloud Foundry, mais sa mise en place est simplifiée. Elle implique en l'occurrence une valeur à ajouter à cf push : –strategy rolling. Et fonctionne aussi avec cf restart (redémarrage d'une app) comme avec cf restage (recréation d'un *droplet*). Il est prévu de la rendre utilisable également avec cf scale (mise à l'échelle) pour éviter les redémarrages.

Kubernetes sur la *roadmap*

Le CLI v7 apporte par ailleurs une commande qui permet de déployer une application à plusieurs processus (par exemple, travail et rendu sur une *web app*). L'idée étant d'avoir un *scaling* indépendant sans tomber dans la complexité des microservices.

Dans la même veine, il y a les processus qu'on appelle, dans le jargon Cloud Foundry, les *sidecars*. Leur prise en charge n'est pas nouvelle, mais les API dont elle dépend sont désormais officiellement prises en charge avec le CLI v7.



Ces processus viennent se greffer dans des conteneurs applicatifs pour partager, par exemple, un système de fichiers, des capacités de montée en charge ou une communication sur socket Unix. Les contrôles d'intégrité sont effectués indépendamment. En revanche, si un processus crashe, l'autre aussi.

On aura aussi noté la possibilité d'ajouter des métadonnées à des objets tels que les applications et les espaces. Qu'il s'agisse d'annotations, d'étiquettes ou de sélecteurs (filtres basés sur des étiquettes, un objectif : aider à répondre à des questions de type « Où en est cette application dans son cycle de vie ? Qui en est propriétaire ? Quel département s'en occupe ? » etc.

On consultera ce document pour assimiler les différences fonctionnelles entre les CLI v6 et v7. Et on gardera un œil sur la feuille de route, où figure notamment la prise en charge de Cloud Foundry for Kubernetes (cf-for-k8s). Pivotal (VMware) emmène ce projet dont la première version est sortie en avril. Il prend la forme d'une distribution du PaaS native à Kubernetes.

Photo d'illustration © Ra2studio – Shutterstock