

DevOps : Canonical pousse Juju avec ou sans Kubernetes

Canonical accentue sa communication à propos de Juju.

Début novembre, Mark Shuttleworth en avait [modifié](#) la description sur GitHub. Il l'avait rendue plus précise. D'« outil DevOps » destiné au pilotage des architectures applicatives, Juju devenait un « gestionnaire de cycle de vie d'opérateurs ».

Les opérateurs en question se sont développés avec l'essor de Kubernetes. Ils font office de contrôleurs « sur mesure » permettant d'automatiser la gestion des applications et de leurs composants. On leur fournit une configuration de haut niveau qu'ils traduisent en actions de bas niveau à partir des bonnes pratiques intégrées à leur logique.

À travers Juju, Canonical a étendu cette approche aux applications non conteneurisées. Elle englobe aujourd'hui des plates-formes de cloud public (AWS, Azure, Google, Oracle), ainsi qu'OpenStack, VMware et les substrats *bare metal*. Un opérateur conçu pour ce type d'environnement* « peut partager 95 % de son code » avec sa variante Kubernetes, nous affirme-t-on.

Open Operator Collection : de la matière pour Juju

Ressource majeure pour Juju : l'Open Operator Collection. Les opérateurs que regroupe cette plateforme communautaire sont distribués sous forme de charmes (représentations de services). On pouvait encore récemment les classer par ordre alphabétique ou antialphabétique. Des catégories « Tous », « Linux » et « Kubernetes » s'y sont substituées.



Canonical a assorti la démarche d'un [post](#) qui officialise l'ouverture de l'Open Operator Collection aux applications « traditionnelles ». L'annonce intervient peu avant la KubeCon + CloudNativeCon North America.

The largest collection of application operators now supports both cloud-native and traditional applications on Windows and Linux. Read our announcement on the Open Operator Collection and new CharmHub here [#KubeCon](#) <https://t.co/QXj81feudu> <https://t.co/gkMpjEQQ1h>

— Ubuntu (@ubuntu) [November 13, 2020](#)

Au-delà du cycle de vie des applications, il s'agit d'assurer leur intégration. L'Open Operator Collection favorise cet aspect en donnant accès à des charmes standardisés, réutilisables et éventuellement *open source*.