

# Kubernetes prend ses distances avec Docker

Ian Coldwater a-t-il eu les bons mots à propos de la fin du support de Docker dans Kubernetes ? Lui-même en doute.

Cet ambassadeur de la Cloud Native Computing Foundation a d'ailleurs présenté ses excuses pour avoir « causé une panique ». La raison : un tweet du 3 décembre, supprimé depuis lors. L'intéressé y adressait un avertissement : « *Faites attention et préparez-vous. Cela va casser vos clusters* ».

Il n'avait pas précisé que la fin du support ne concernait que la partie *runtime* de Docker. Un [billet](#) publié la veille sur le blog officiel du projet Kubernetes l'explique. Principal message : « *Ce n'est pas aussi dramatique que ça en a l'air* ».

Que se passe-t-il en pratique ? Dans les grandes lignes, le changement est lié à l'**arrêt de la prise en charge de dockershim dans le kubelet**. Ce dernier s'exécute sur chaque nœud d'un cluster pour s'assurer du bon fonctionnement des conteneurs. Utilisé avec le *runtime* Docker, il requiert un module (shim) pour implémenter l'interface standard CRI (Container Runtime Interface, qui démarre et arrête les conteneurs). C'est donc un outil de plus à maintenir pour la communauté Kubernetes.

Celle-ci a décidé d'encourager l'adoption de moteurs d'exécution qui n'ont pas besoin d'un tel *plugin*. Elle a franchi une étape dans ce sens avec la sortie de la version 1.20 de l'orchestrateur : Docker est désormais [officiellement obsolète](#). En première ligne pour le remplacer, un *runtime* sur lequel il s'appuie : containerd.

## Développeurs, vous pouvez garder Docker

De manière générale, **rien ne changera pour les développeurs**. Les images qu'ils concevront avec Docker continueront à fonctionner. C'est du côté des admins qu'on sera attentif.

Sur les versions managées de Kubernetes, il s'agira de contrôler que les *workers* (nœuds chargés d'exécuter les applications) utilisent bien un *runtime* pris en charge. AWS, par exemple, a déjà promu containerd [par défaut](#).

Pour ceux qui déploient eux-mêmes leurs clusters, il faudra éventuellement adapter les configurations. Notamment au niveau du provisionnement, de la journalisation ou encore de la connexion à des ressources de type GPU (plus de détails dans la [FAQ](#)).

Pour le moment, avec Kubernetes 1.20, seul un message d'avertissement s'affiche dans kubelet. La fin effective du support de Docker est prévue au plus tôt pour fin 2021 avec la version 1.23.

*I'm sorry I caused panic this morning. I wanted to call attention to the deprecation of Docker support in advance so people could make a plan for it, because for some it will be a major breaking change that will take time to fix.*

*It, uh, worked? But it also stressed people out.*

— Ian Coldwater   (@IanColdwater) [December 3, 2020](#)

*Illustration principale © Docker*