

# CoreOS lance sa Rocket 1.0 contre Docker

En décembre 2014, CoreOS, éditeur d'un système d'exploitation pour les datacenters, lançait une petite bombe ou plus exactement une fusée dans la communauté des conteneurs Linux. La start-up avait présenté son projet, [Rocket \(rkt\)](#), une alternative à la dominance de Docker. Les équipes de CoreOS ont travaillé depuis la version 0.1 de l'outil de gestion des conteneurs LXC et viennent de lancer officiellement la mouture 1.0 de rkt avec comme ambition de fournir les mêmes fonctionnalités que Docker, remplir les mêmes tâches et dans certains cas être complémentaires.

Par rapport à la version bêta de rkt 1.0, le CTO de CoreOS, Alex Polvi souligne dans un blog que les changements sont minimes en renforçant les interfaces et les API de rkt. Un effort particulier a été mené sur la sécurité en prenant en considération les meilleures pratiques qui existent dans le monde Unix. Cela passe par exemple par la signature et la validation des images, la séparation des privilèges (certains composants peuvent avoir l'autorisation d'accéder à Internet, mais pas avec des droits administrateurs), etc.

## Un lancement sous les auspices de l'Open Container Project

Cet effort sur la sécurité est clairement fait pour séduire les développeurs qui utilisent déjà Docker où la sécurité est souvent considérée comme son talon d'Achille. Une vision qui est en train de se résorber avec plusieurs annonces qui vont dans le bon sens.

Pour autant, si il y a une guerre pacifique entre CoreOS et Docker, les deux start-ups n'oublient pas que dans le monde Open Source, la question de l'interopérabilité est essentielle. Rocket s'appuie sur un format d'image pour les conteneurs, APCC (Application Container Spec ) pendant que Docker délivre en OCI.

CoreOS assure la portabilité d'un conteneur sur les deux runtimes (cf schéma ci-dessous). Une exigence actée lors de la signature d'une paix des braves sous le label [Open Container Project](#). L'objectif du consortium est de développer un standard commun, qui devra être rétro-compatible avec Docker, indépendant de tout client et outil d'orchestration ainsi que portable sur de multiples plates-formes (Linux, Windows, Solaris, architectures Power et x86).

## OCI and appc Intersections

	Standards		Container Runtimes		Portability
	OCI	APPC	DOCKER	RKT	EXAMPLES
Container Image	✗	☑	Docker Format v2	appc Image Spec Docker Format v2	User builds container once, can run in docker or rkt
Image Distribution	✗	☑	Docker Registry Protocol	appc Discovery Spec Docker Registry Protocol	Docker and rkt can share container registry mechanism
Runtime	☑	☑	libcontainer	appc Runtime Spec	Docker and rkt can share exec drivers (impl. using LXC, runc, systemd-nspawn, etc)
On-disk Image Format	☑	✗	Previously unspecified	Previously unspecified	Exec drivers share a metadata format and filesystem layout for how to execute a container

### A lire aussi :

[Avec Tectonic, CoreOS gagne Kubernetes et des sous de Google](#)

[Chef, CoreOS, Jenkins et les autres : la boîte à outils Open Source du Cloud](#)

[CoreOS se paye Quay.io, un référentiel privé pour les conteneurs Docker](#)