

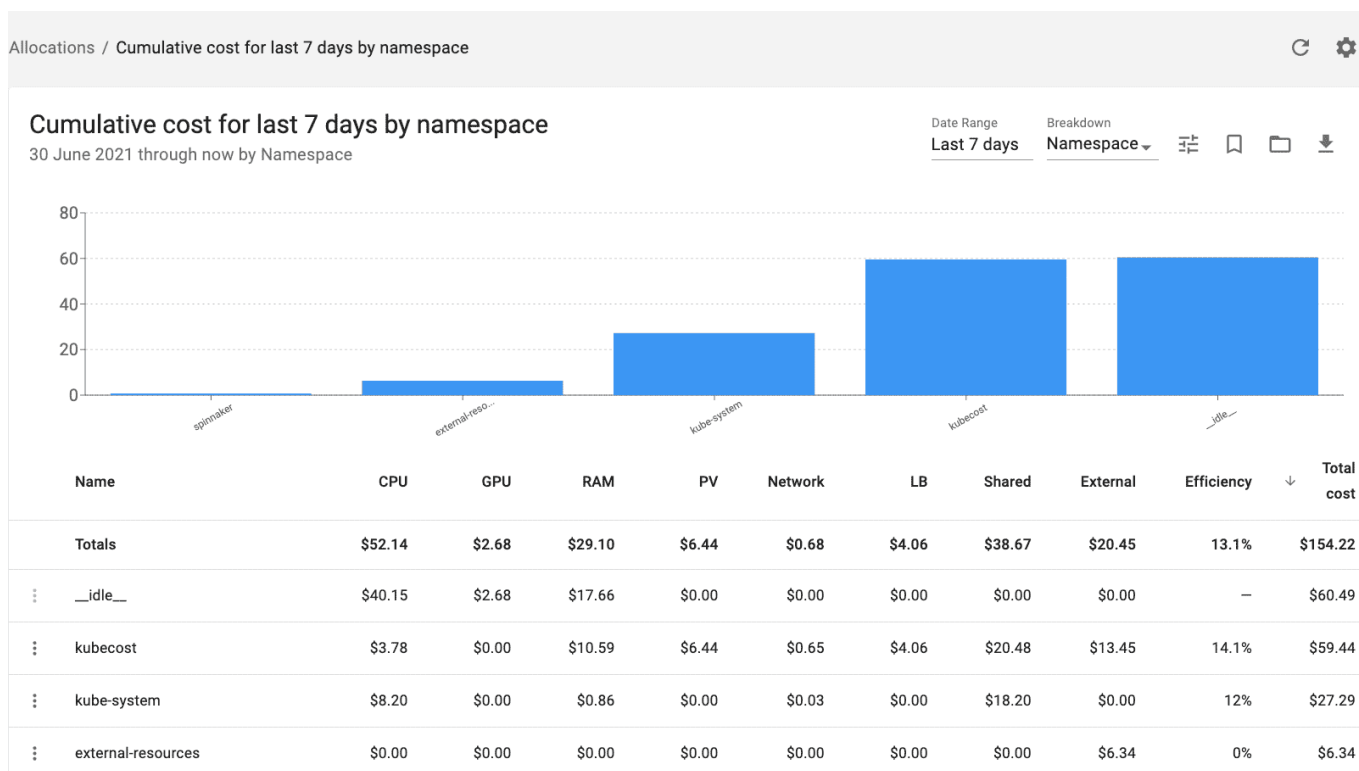
OpenCost : vers un standard FinOps pour Kubernetes ?

Tient-on un standard FinOps pour Kubernetes ? Avec OpenCost, la voie est ouverte. [Le projet](#) vient de se [porter candidat](#) à la CNCF (Cloud Native Computing Foundation). Avec le soutien de poids lourds dont Adobe, AWS, Google Cloud et SUSE.

À la racine, il y a une entreprise américaine : Stackwatch. Et un produit, lancé en 2019 sur le modèle *open core* : KubeCost. Son pilier : un modèle de mesure et d'allocation des coûts dans les environnements Kubernetes. Et, par-dessus, divers niveaux de service (gratuit, Business, Enterprise).

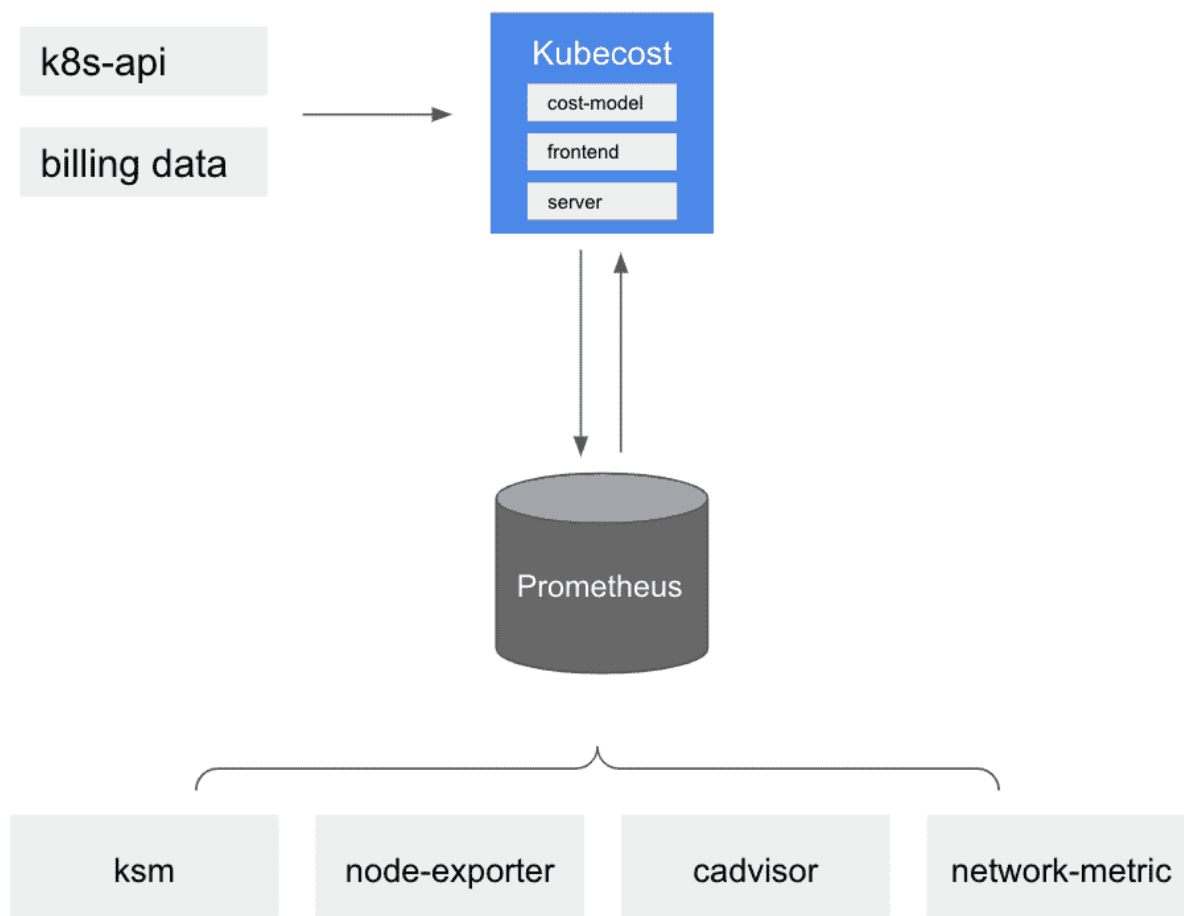
Le projet OpenCost consiste en [ce modèle](#) publié sous licence Apache 2.0 et assorti d'une implémentation en Go. La communauté y a greffé diverses extensions (*templates* CloudFormation, *plug-in* pour l'IDE Lens K8s...) et les forks [se comptent](#) par centaines.

Principale cible d'OpenCost : les déploiements chez les « trois grands » du cloud (AWS, Azure, GCP). Pour chacun, il s'appuie en standard sur les API de facturation. Pour aller plus loin (prise en compte des remises négociées, des instances préemptives, gestion des coûts hors cluster, etc.), on peut y connecter des comptes. Et, pour les déploiements sur site, apporter ses propres données de tarification.



Le cœur fonctionnel d'OpenCost comprend un *front-end* Nginx (avec modèle et un serveur API) et un *back-end* Prometheus. On peut y adjoindre un serveur Grafana et un daemonset pour collecter les métriques réseau. Sur la roadmap figurent notamment l'ajout de test de conformité et l'affichage

des émissions de CO₂. Et la stabilisation de fonctionnalités comme la mise en œuvre automatisée du redimensionnement de ressources.



Certains éléments ne sont accessibles qu'avec des licences payantes de Kubecost : multicluster, extension des périodes de conservation, alertes, SSO... Cadence actuelle du projet : hors correctifs, une mise à jour par mois.

Illustration principale © everythingpossible – Fotolia