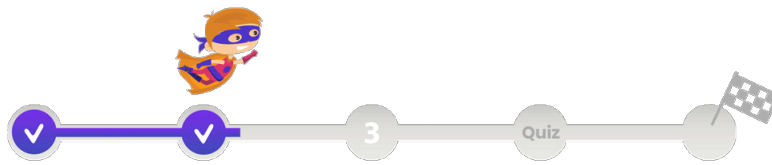


### 3 - Les Orchestrateurs de conteneurs



## 3. Les Orchestrateurs de conteneurs

Maintenant que vous avez des conteneurs qui tournent sur vos serveurs et que vous avez compris les avantages des conteneurs par rapport au VM, il va falloir se pencher sur la vision opérationnelle du DevOps avec la gestion des **conteneurs en production**.



Pour cela, il n'y a **qu'une solution** : utiliser un **orchestrateur de conteneur**.

Dans le cas de Docker, plusieurs choix s'offrent à nous.



En informatique, l'**orchestration** permet de définir un **processus automatisé de coordination** et de **gestion de systèmes complexes**. Il définit les politiques et les niveaux de service grâce à des **flux de travail automatisés**.

Cela crée une infrastructure permettant d'**augmenter** ou de **diminuer** le nombre d'instances d'un serveur en fonction des besoins de chaque application.

L'orchestration assure également la **gestion centralisée du pool de ressources**, y compris pour la facturation. Le besoin de ressources supplémentaires ou de nouvelles applications est déclenché par des **outils automatisés** pouvant exécuter des tâches qui auparavant ne pouvaient être effectuées que par une ou plusieurs personnes.

### 3.1 Rancher et Cattle

« **Rancher** », à ne pas confondre avec « RancherOS », est un **méta-orchestrateur de conteneurs** développé par l'entreprise américaine « Rancher Labs ».



Figure 1 : Logo Rancher

**Mais vous allez me demander, un méta-orchestrateur, c'est quoi ?**



Un **méta-orchestrateur**, est un logiciel permettant de **gérer plusieurs orchestrateurs**.

Rancher permet donc de **gérer plusieurs orchestrateurs** tels que « Swarm », « Kubernetes » ou « Mesos ». Ils ont cependant un autre orchestrateur maison nommé « Cattle » qui ne peut être déployé qu'avec Rancher.

management			10 Services	63 Containers	
Active	cattle	Image: rancher/none	Service	6 Containers	
Active	go-machine-service	Image: rancher/server:v1.0.1	Service	2 Containers	
~ Active	load-balancer	To: cattle ,websocket-proxy Ports: 80 , 443 , 81 , 444	Load Balancer	3 Containers	
~ Active	load-balancer-swarm	To: websocket-proxy-ssl Ports: 2376/tcp	Load Balancer	3 Containers	
Active	rancher-compose-executor	Image: rancher/server:v1.0.1	Service	2 Containers	
Active	redis	Image: rancher/none	Service	3 Containers	
Active	tunnel	Image: rancher/none	Service	37 Containers	
Active	websocket-proxy	Image: rancher/server:v1.0.1	Service	2 Containers	
Active	websocket-proxy-ssl	Image: rancher/server:v1.0.1	Service	2 Containers	
Active	zookeeper	Image: rancher/none	Service	3 Containers	

Figure 2 : Interface graphique de Rancher 1.6

L'avantage principal de Rancher vient de son **interface utilisateur**. Elle permet à toute personne débutant avec Docker de facilement prendre en main l'orchestration de conteneur.

Toutes les options de la commande « docker run » sont présentes lors de la création d'un conteneur.

Il existe une **couche réseau IPSEC supplémentaire** permettant de faire communiquer tous les hôtes faisant tourner Docker de façon à créer un cluster Docker.

Parmi les **fonctionnalités principales de Rancher**, on retrouve :

- Un load-balancer
- Des mécanismes de retour arrière (Rollback)
- Des mécanismes de mise à jour progressive (Rolling Upgrade)
- Un service DNS de résolution de nom entre les services
- Un système de catalogue pour faciliter l'installation de service
- Gestion de secret
- Gestion des drivers de stockage

## 3.2 Kubernetes

Il y a les entreprises qui arrivent à innover sur de courtes périodes et qui arrivent à sortir un nouveau produit juste avant les autres comme c'est le cas avec Netflix et Facebook.

Ensuite, nous avons les entreprises qui font comme beaucoup d'autres, mais qui le font mieux, c'est le cas de Amazon ou Apple.

Mais dans le monde de l'informatique, il y a surtout une entreprise qui sait être visionnaire sur 7 à 8 ans. Le **géant Google** a une fois de plus montré son avancée technologique. Quelques années après le billet de recherche sur Hadoop qui avait vu naître l'ère du BigData, Google revient en force avec une nouvelle technologie : Kubernetes.

Vous trouverez la publication ici : <https://research.google.com/pubs/pub44843.html>.

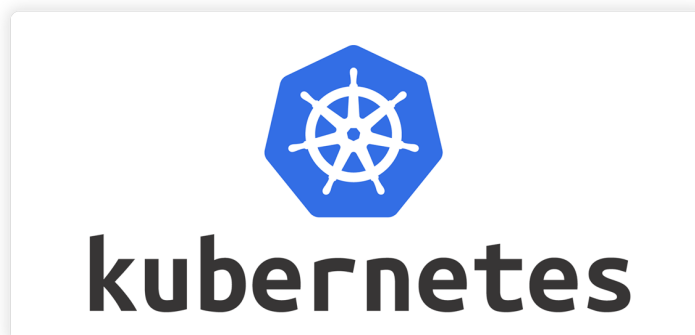


Figure 3 : Logo de Kubernetes

Le monde entier a commencé à utiliser les conteneurs massivement avec l'arrivée de Docker en 2013-2014.

Mais en **2014**, **Google** libère les sources de la troisième version de son **orchestrateur maison**. Initialement nommé Borg puis Omega, pour les premières versions, c'est donc Kubernetes, communément appelé « K8s », qui arrive dans nos mains en 2015 pour la version 1.0.

Google décide alors de **libérer les sources de son orchestrateur** via un partenariat avec la « Linux Foundation » afin de créer la « **Cloud Native Computing Foundation** » (CNCF). Cette dernière devient un incubateur pour beaucoup de projets liés qui gravitent autour du conteneur.

Vous trouverez la liste actuelle des projets sur <https://www.cncf.io/projects/>.

**Kubernetes est le nec plus ultra des orchestrateurs**. En moins de 3 ans, il a évolué de 10 versions majeures, soit presque quatre mises à jour par an.

Avec plus de 1600 contributeurs au projet, l'orchestrateur se voit étoffé de fonctionnalités de plus en plus complexes. Il

permet depuis peu de gérer une granularité de droits très fine en suivant le modèle RBAC (Role-Based Access Control).



Malheureusement, il reste très **lourd à déployer** et demande de **maîtriser des notions assez complexes**. C'est pourquoi vous trouverez de plus en plus de clouds provider qui vous proposeront une offre « Kubernetes as a Service » permettant de ne s'occuper que de la partie logicielle.

Le succès de K8s est, comme pour Docker, dû à son API et à son CLI qui permettent de **piloter l'intégralité des services**. Le concept va encore plus loin, car il est possible d'enrichir l'API de Kubernetes en **consommant l'API**.

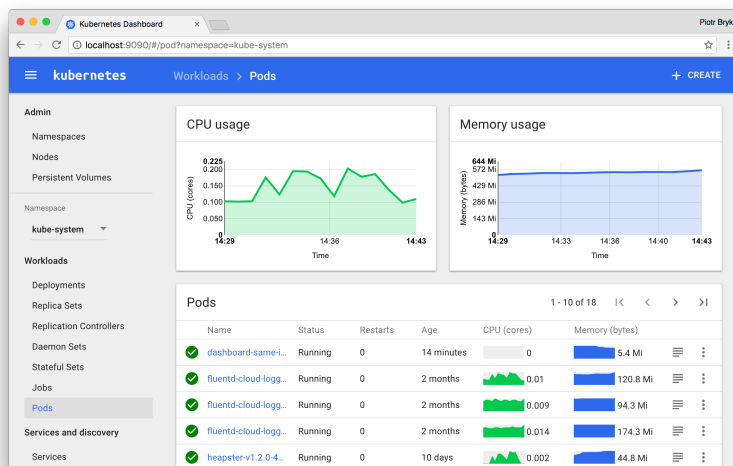


Figure 4 : Dashboard de Kubernetes

Contrairement à Rancher, l'interface graphique de Kubernetes est très austère, elle permet surtout de faire de la **visualisation d'information** sur l'état du cluster, plus que du contrôle.

### 3.3 OpenShift



Figure 5 : Logo d'OpenShift



OpenShift est la solution de PaaS de Red Hat. C'est une offre de **Cloud public et privé** qui est **orientée sur les conteneurs**.

Ce n'est pas vraiment un orchestrateur, car on y trouve plus de fonctionnalité avec par exemple une **chaîne d'intégration continue**.

Je n'en parlerai pas longtemps, mais il faut savoir que Red Hat a changé son moteur d'orchestration il y a quelques temps afin d'utiliser Kubernetes.

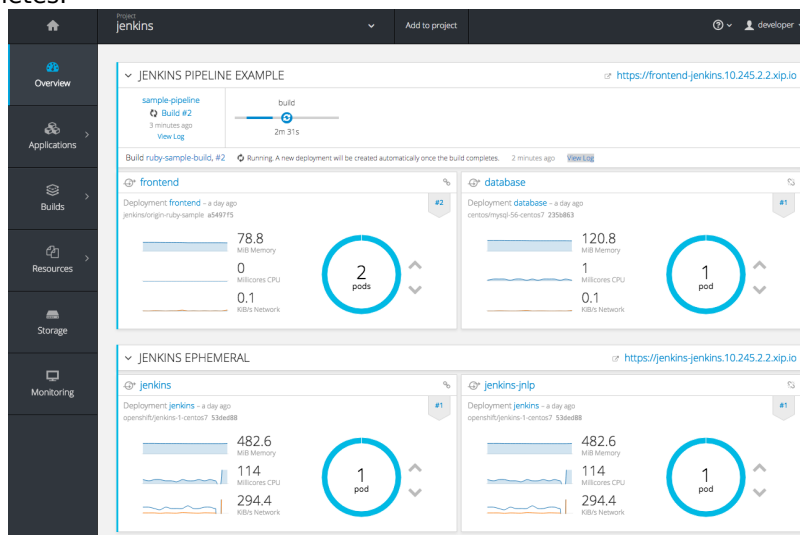


Figure 6 : Dashboard d'OpenShift

Au mois de Juin de 2016, on a vu fleurir un bon nombre d'articles intitulés « la guerre des orchestrateurs arrive », car Docker venait d'intégrer son propre orchestrateur nommé Swarm au sein du moteur de Docker. La bataille fut de courte durée : un peu plus d'un an plus tard, en octobre 2017, Docker annonce officiellement avoir intégré Kubernetes dans sa plateforme de service Entreprise.

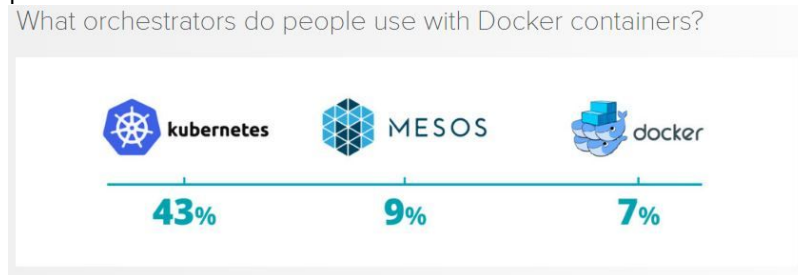


Figure 7 : Orchestrateur le plus utilisé en 2017 selon Sysdig



Pour voir la fiche complète et les documents attachés, rendez-vous sur <https://elearning.26academy.com/course/play/5ac4f38a5bdb48ce2758cf>