

Se simplifier Kubernetes grâce à Azure ? Merci AKS !



kubernetes



Kubernetes

Introduction à Kubernetes

D'abord il y eu les conteneurs :

- Docker permet de créer et exécuter facilement nos conteneurs

Mais

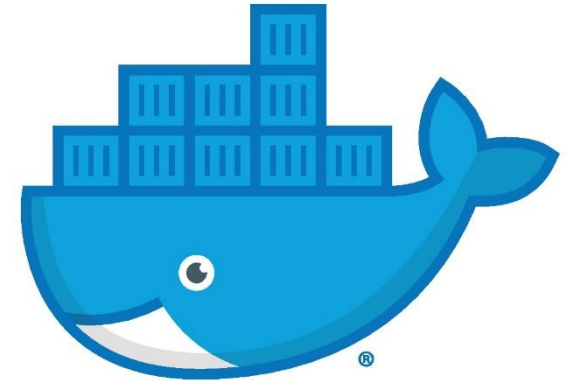
- Comment gérer et planifier leur cycle de vie ?
- Comment monter en charge ?
- Comment manager une grosse quantité de container ?

Puis les orchestrateurs

- K8s : réécriture open source de Borg
- Compatible on premise et Cloud
- Maintenu par Google, Microsoft, RedHat

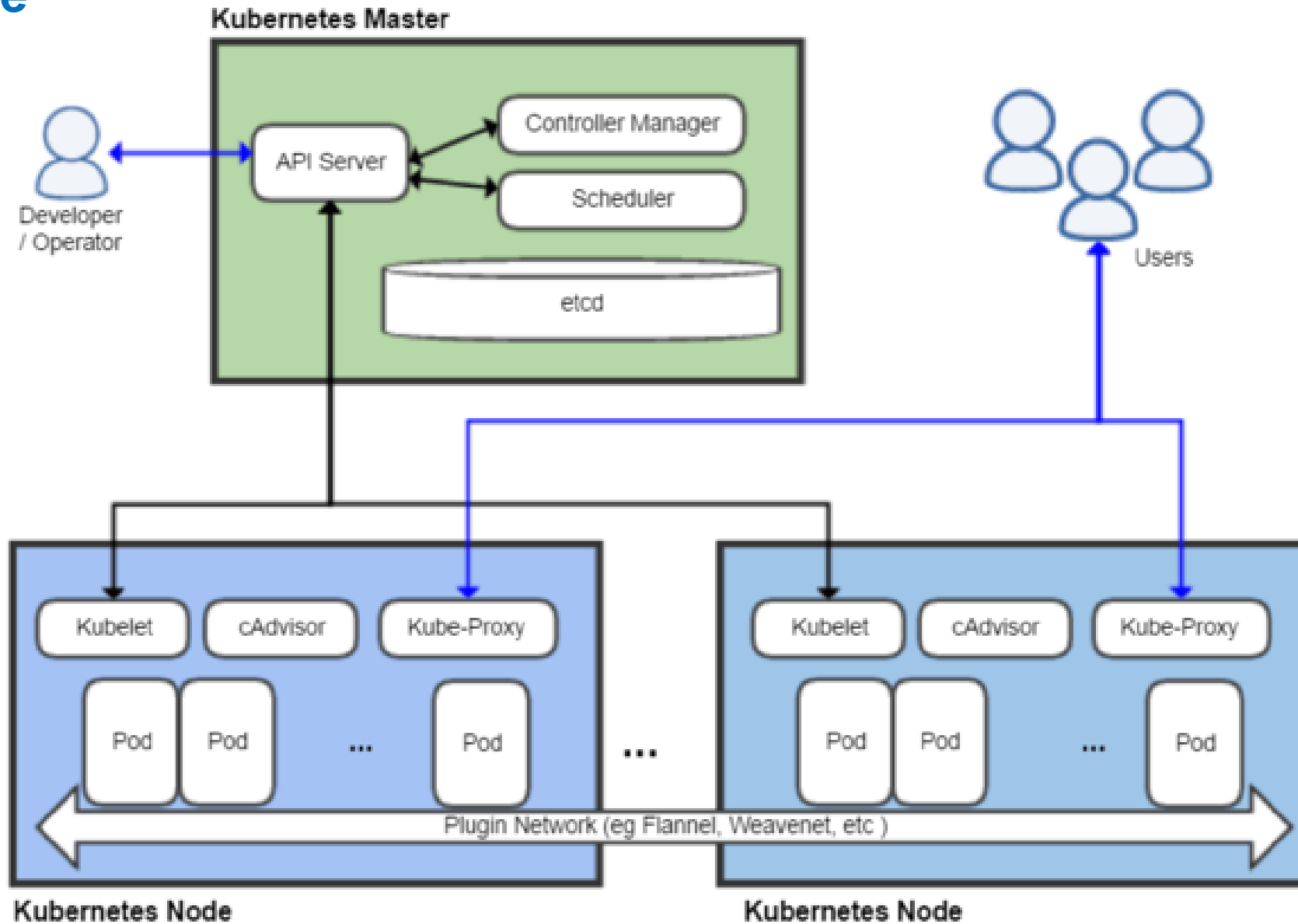
Que permet de faire Kubernetes ?

- Automatiser le déploiement et la réplication de conteneurs
- Haute disponibilité
- Loadbalancing
- Détection de problème
- Définition d'un état attendu



kubernetes

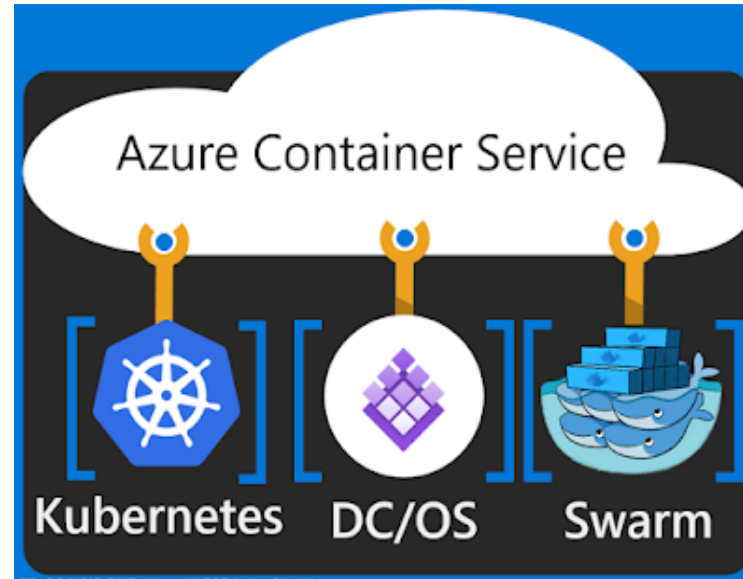
Architecture



Azure

Orchestrateur

- Offre managée :
Azure Kubernetes Services
 - Haute disponibilité
 - Scalabilité à grande échelle
 - Gestion des mises à jour
 - Monitoring
-
- SLA des nodes : 99.9%
 - Availability Set => 99.95%
 - Availability Zone => 99.99%



Démo

Pour aller plus loin

■ Fonctionnalités k8s

- LivenessProbe : vérifie si le container est fonctionnel
- ReadinessProbe : détermine quand le container est prêt à recevoir du trafic
- Ressourcequota : restreint l'utilisation des ressources au niveau namespace.
Utilise si plusieurs équipes ou environnements différentes

- **Istio** : surcouche a k8s (pattern façade) : https, métrique des appels, échec, cartographie du SI, blue/green deployment (règle de routage)



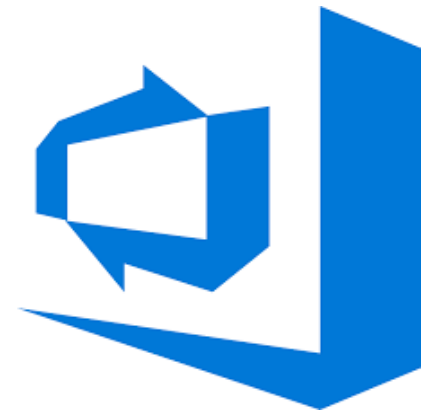
■ Monitoring

- Prometheus
- Graphana



Pour aller plus loin

- **Cluster autoscale** : via VMSS
 - az aks update --resource-group myResourceGroup
 - name myAKSCluster --update-cluster-autoscaler --min-count 1
 - max-count 5
- **Tests**
 - K8s local : intégré nativement au client Docker
 - AKS DevSpaces : port forwarding sur le cluster pour debugger
- **Sécurité**
 - RBAC / AD
 - Pod policies
- **Réseau**
 - Azure APIM
 - Azure Firewall
 - VNET
- **Déploiement**
 - Helm
 - Azure DevOps



Merci