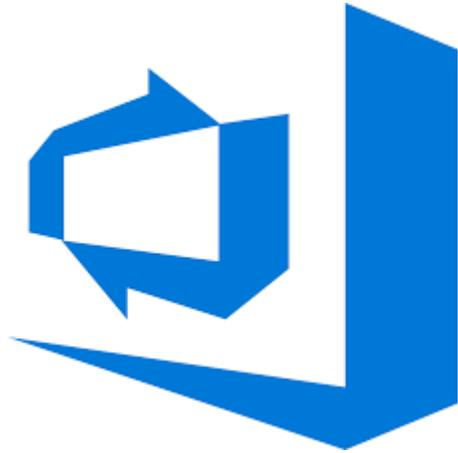


Du DevOps multi Cloud avec Kubernetes et terraform



Google Cloud Platform



kubernetes



Sébastien Meynier : GFI

Yves de Caqueray : GFI

Thomas Rannou : Sopra Steria

thomasrannou.azurewebsites.net

 @thomas_rannou

Du DevOps multi Cloud avec Kubernetes et terraform

Pourquoi ce talk ?

Peut-on provisionner un cluster Kubernetes et y déployer une application tout en restant agnostique de la plateforme Cloud utilisée ?

75% des entreprises envisagent le multicloud à court terme.

Mots clés :

Kubernetes, Terraform, Azure DevOps, Azure, AWS, GCP



Kubernetes

Introduction à Kubernetes

D'abord il y eu les conteneurs :

- Docker permet de créer et exécuter facilement nos conteneurs

Mais

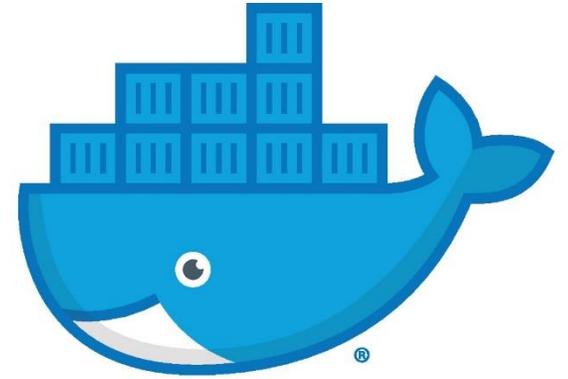
- Comment gérer et planifier leur cycle de vie ?
- Comment monter en charge ?
- Comment manager une grosse quantité de container ?

Puis les orchestrateurs

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

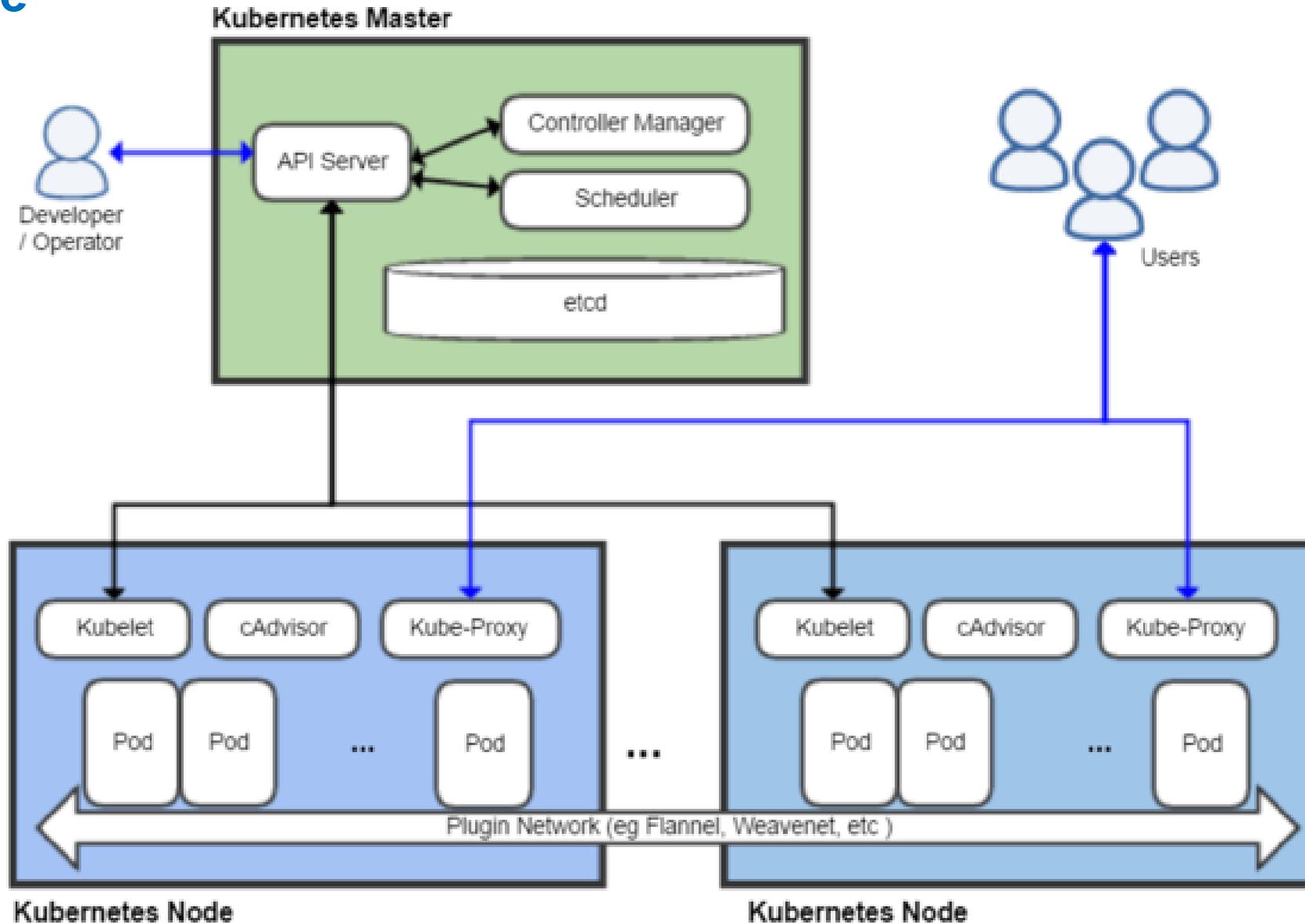
Que permet de faire Kubernetes ?

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



kubernetes

Architecture



KUBERNETES

KUBERNETES EVERYWHERE

Orchestrateur

- Offre managée des providers :

Azure Kubernetes Services

Amazon EKS

GKE

- Haute disponibilité
- Scalabilité à grande échelle
- Gestion des mises à jour
- Monitoring



Google Cloud Platform

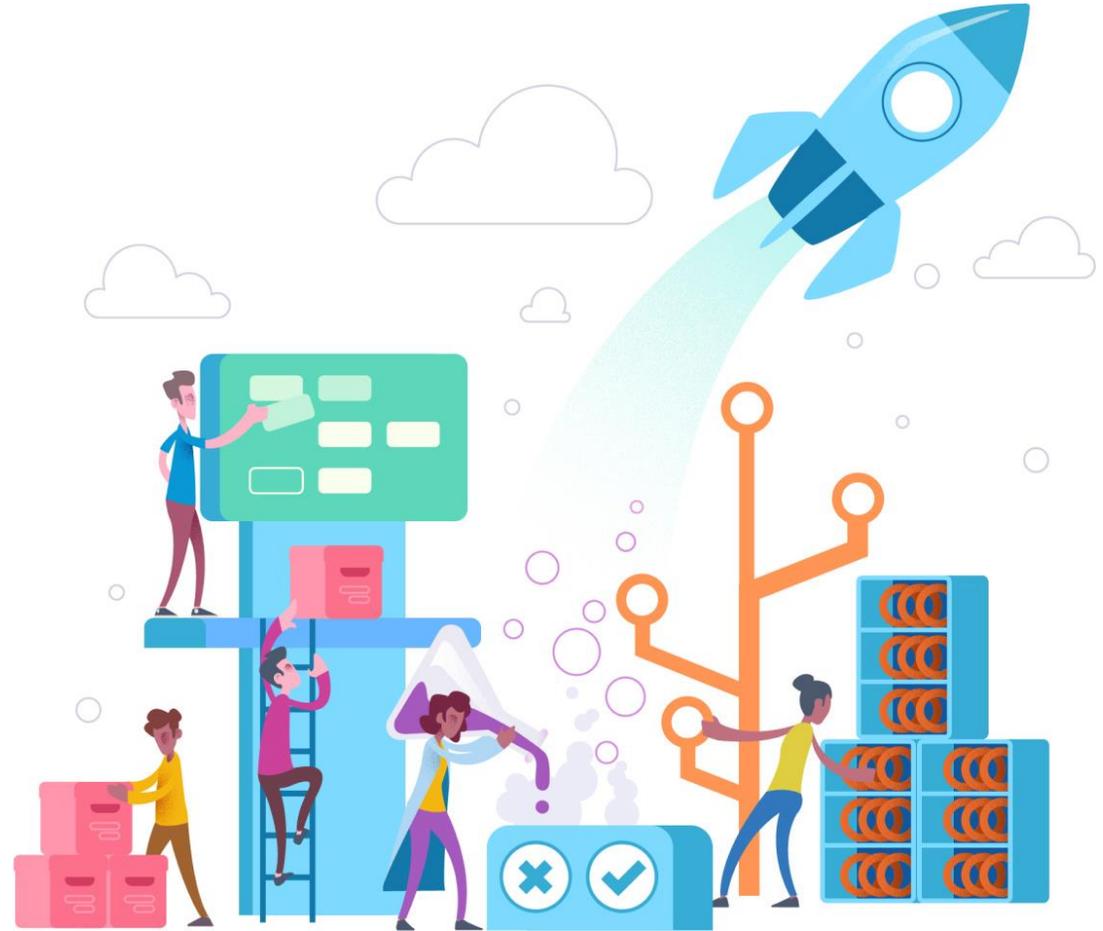


Infrastructure As Code



Azure DevOps

Any language, any cloud, any platform



Démo

Timeline

- **Infrastructure** : 3 repos Azure DevOps
 - I – Présentation du code Terraform
 - II – Intégration et déploiement continu du code Terraform
- **Application** : 1 repo Azure DevOps
 - I – Push sur DockerHub avec un pipeline commun
 - II – Configuration des clusters
 - III – Création des services connexion
 - IV – Intégration et déploiement continu

Conclusion

Pourquoi ce talk ?

Peut-on provisionner un cluster Kubernetes et y déployer une application tout en restant agnostique de la plateforme Cloud utilisée ?

Ce qu'il faut retenir ?

1) L'adhérence

- Forte au niveau de l'infrastructure.
- Mais très peu pour le pipeline d'automatisation.
- Quasiment nulle pour le deploy grâce a k8s

Partie spécifique

- Le script Terraform

Partie commune

- Le pipeline de déploiement du Terraform
- Mon application et son Dockerfile
- Pipeline de déploiement applicatif

- Cout de déploiement multicloud : adapter le terraform.

2) Code everything

- On code son application
- On code son infrastructure
- On code son intégration / déploiement continu

3) Un code Terraform agnostique, est-ce possible ?

- Des notions similaires
- Pourquoi pas ?

Merci