

# Orchestration et automatisation

ESIR 3 – 2021/2022

Djob Mvondo

# Déploiement ???

Automatisation

Choix des nœuds

En continu

Surveiller

Automatiser l'installation des dépendances, des binaires, et le paramétrage de l'application/module.

# Déploiement ???

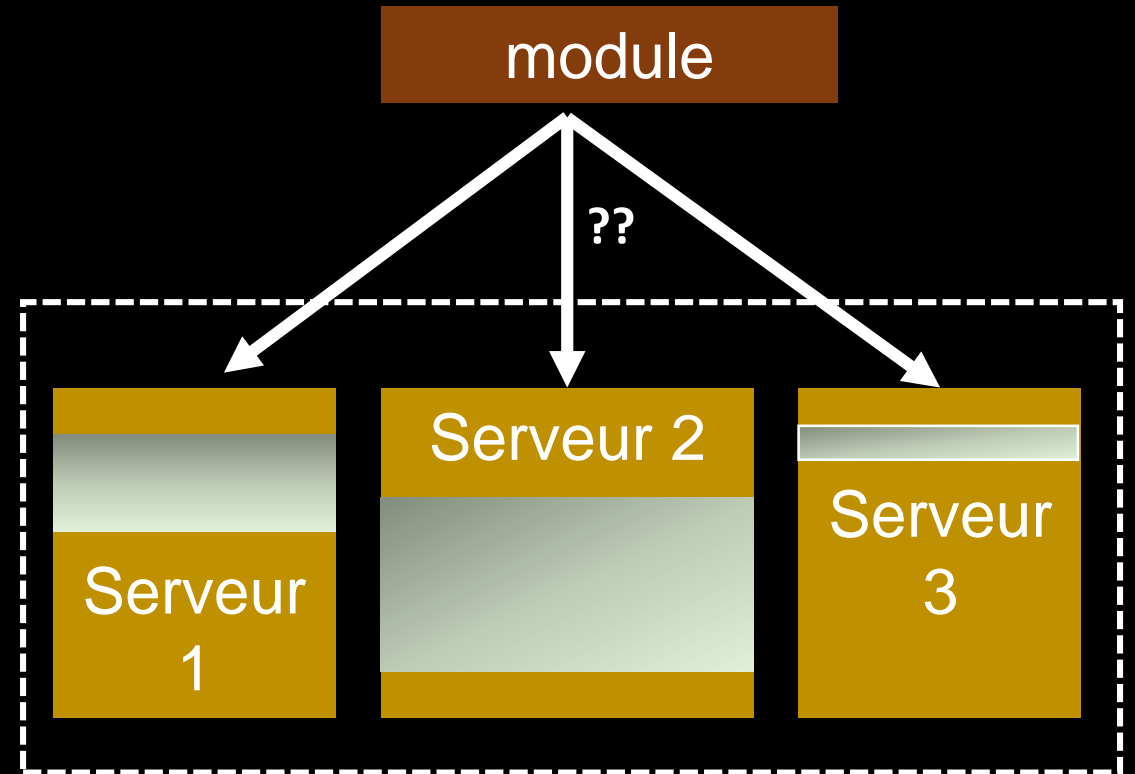
Automatisation

Choix des nœuds

En continu

Surveiller

 Taux utilisation  
ressources



Quel serveur choisir ?  
(Bin packing problem)

# Déploiement ???

Automatisation

Choix des nœuds

**En continu**

Surveiller

Mettre à jour un module en cours  
d'exécution  
(en limitant le temps d'indisponibilité)

# Déploiement ???

Automatisation

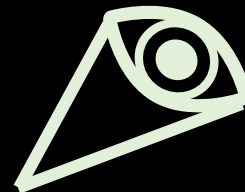
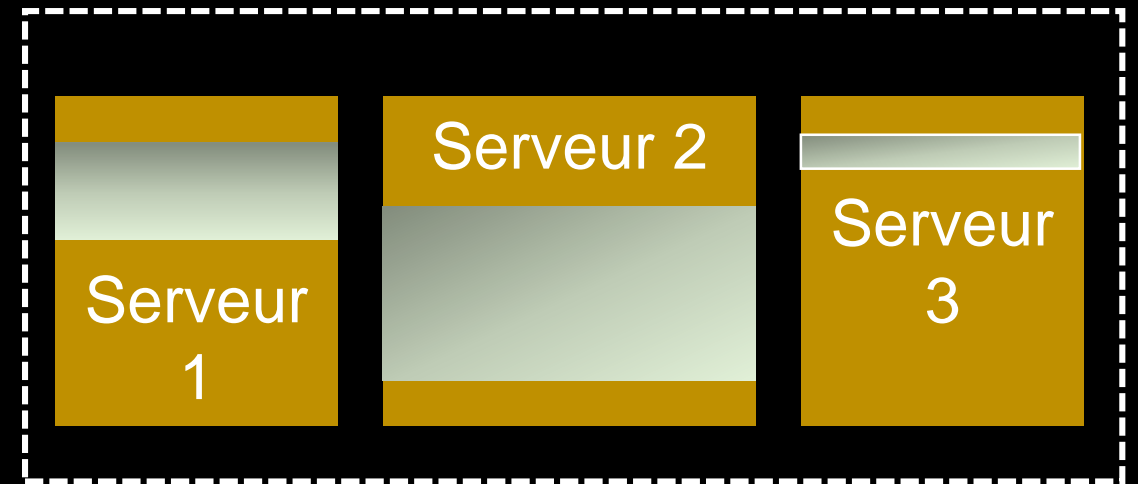
Choix des nœuds

En continu

**Surveiller**



Taux utilisation  
ressources



Observer les modules et  
déclencher les actions  
correspondantes



# Automatisation – docker-compose

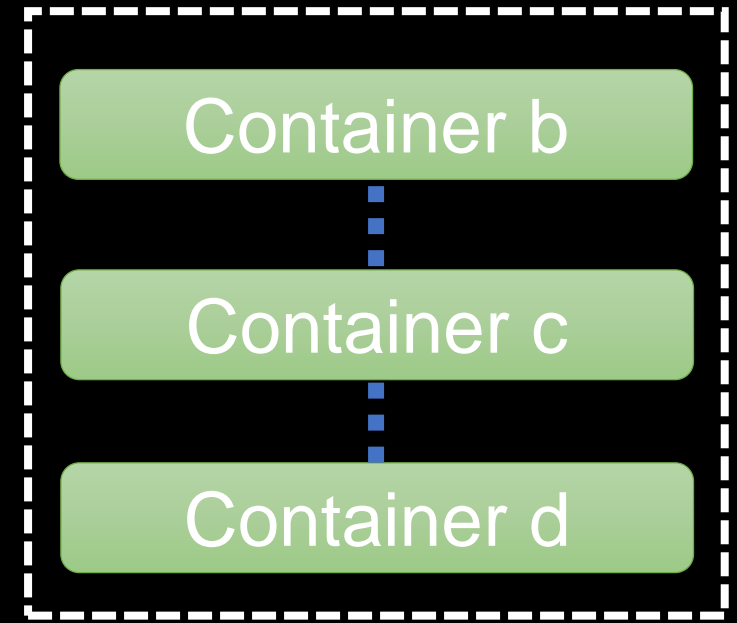
**docker-compose** permet l'automatisation le déploiement d'une application multi-container --- en automatisant les différentes configuration à effectuer



*.yaml*



*docker-compose*



# Déploiement ???

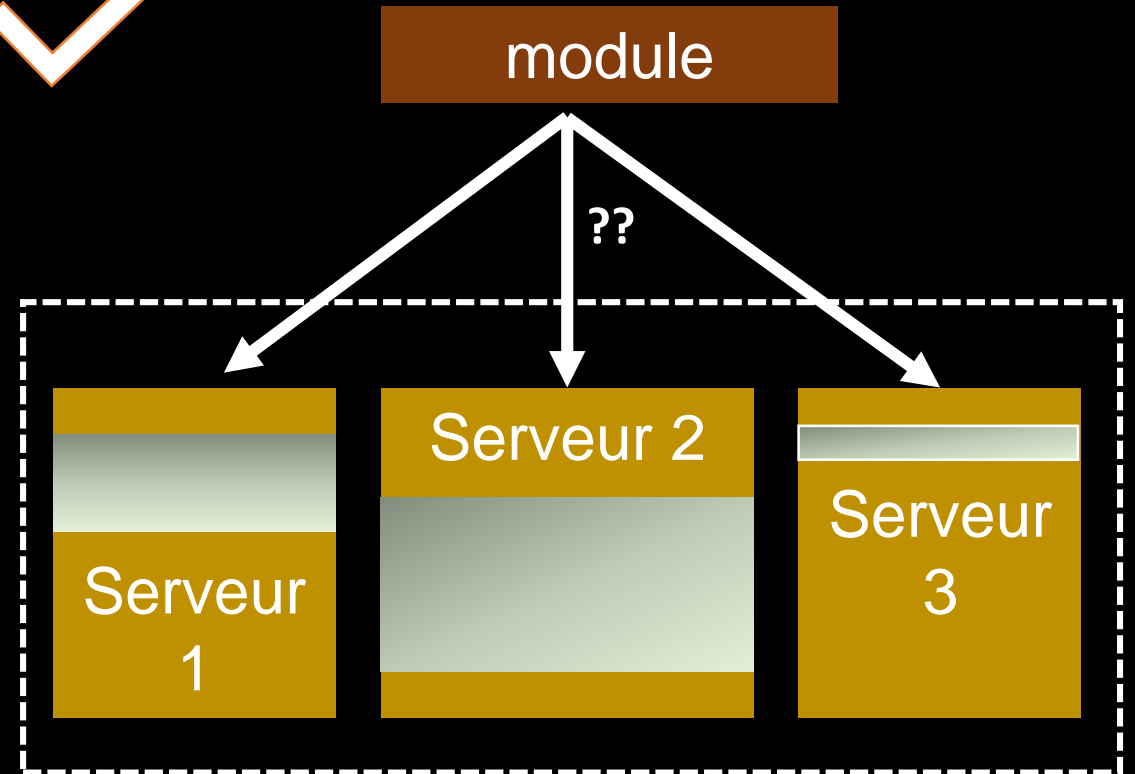
Automatisation

Choix des nœuds

En continu

Surveiller

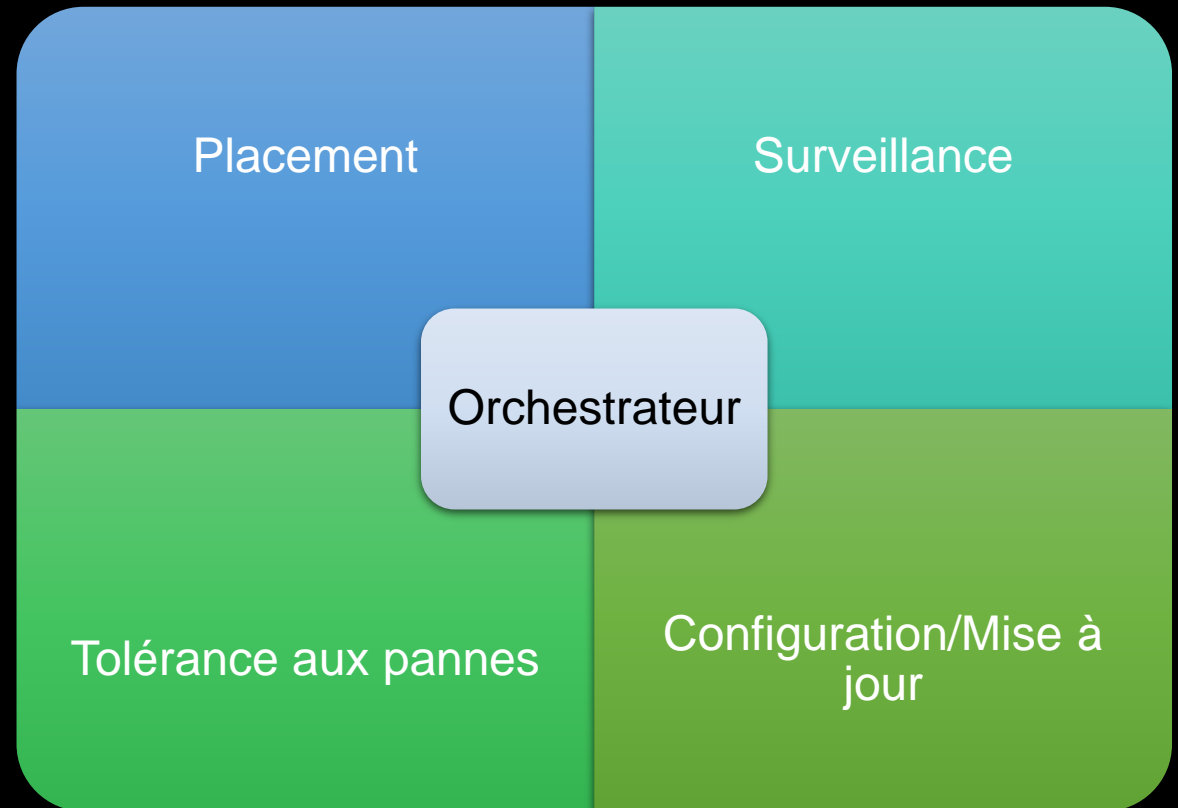
 Taux utilisation  
ressources



# Orchestrateur

L'orchestrateur se charge de la **gestion** des unités d'isolations.

Elle gère le **placement** des unités sur les différents nœuds en fonction du **taux d'utilisation** et **surveille** les unités pour assurer leur **disponibilités**.

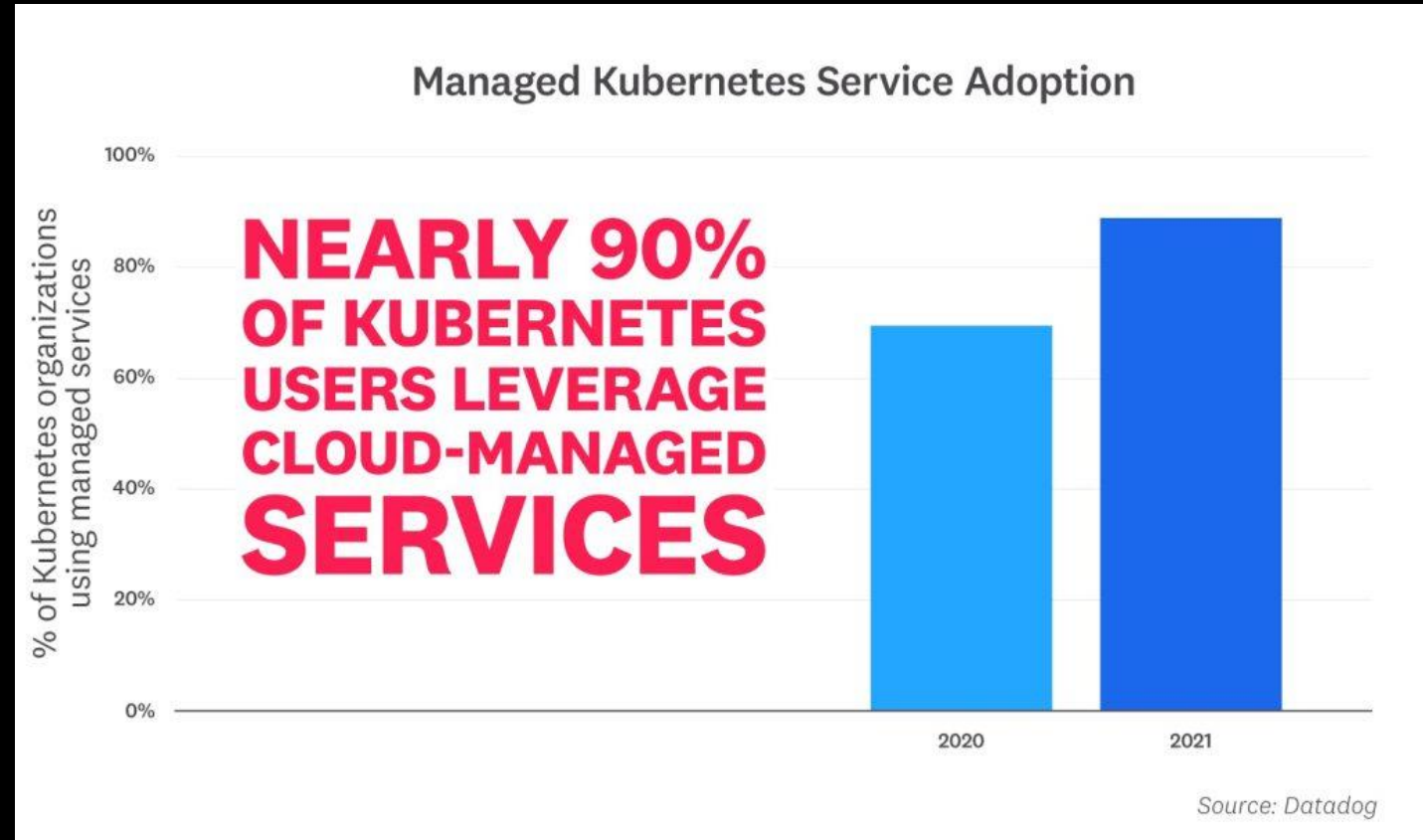




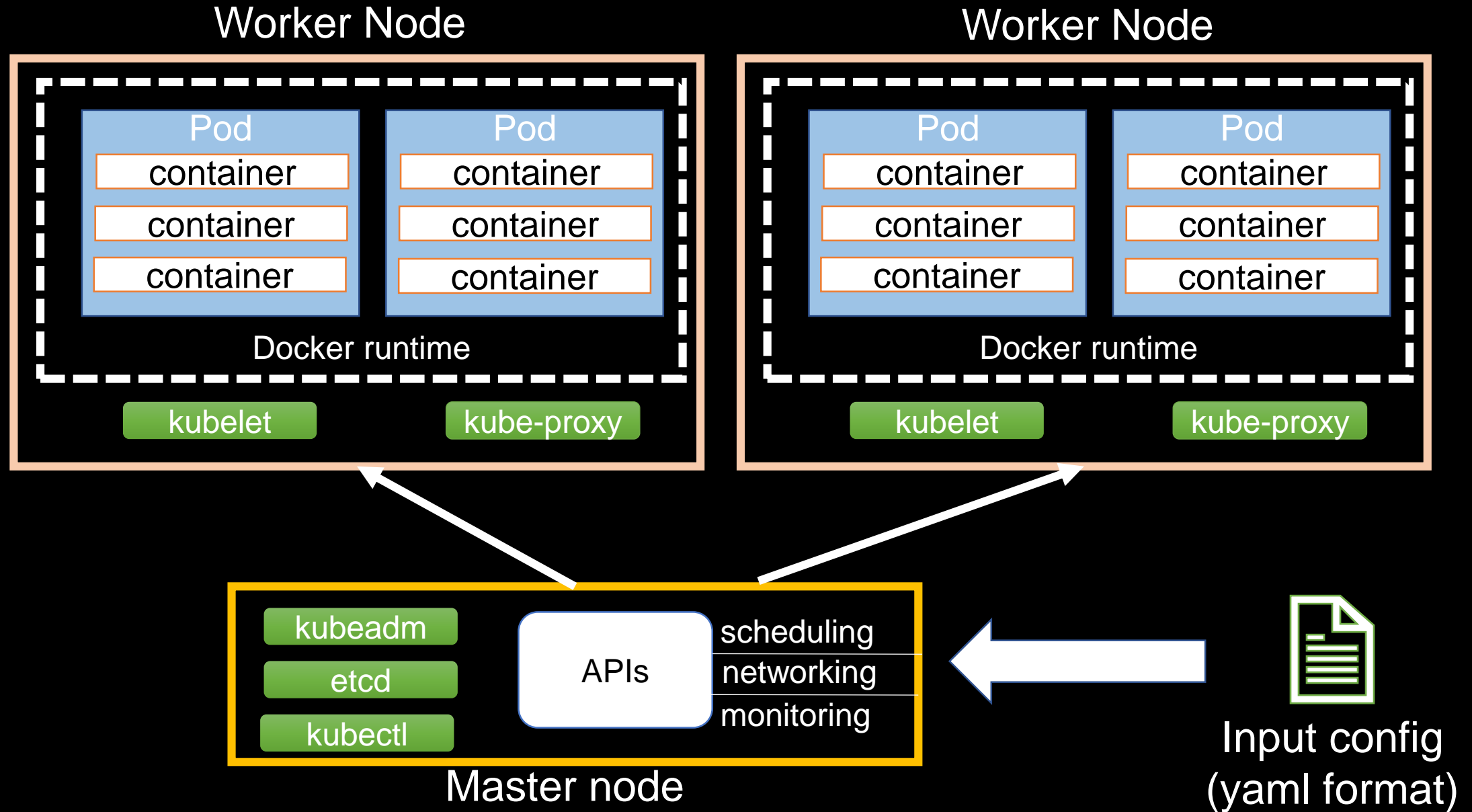
# Orchestrateur - Containers

Plusieurs orchestrateurs open-source et propriétaires existent pour les containers, Docker Swarm, Openshift, Kubernetes, Mesos, Nomad, Rancher, etc.

Nous allons illustrer les concepts d'un orchestrateur avec **Kubernetes**, mais tous reposent sur les même principes.



# Orchestrateur – Kubernetes (k8s)



# Orchestrateur – Kubernetes (k8s)

Kubernetes repose sur une architecture master/workers.

Les nœuds workers possède l'environnement pour l'exécution des containers.

Le nœud master communique avec les workers pour surveiller l'état des containers grâce au service **kubelet** et persiste les données sur **etcd**.

Les nœuds workers créent des containers dans une abstraction appelé **pods** qui symbolise un groupe de containers qui réalisent un objectif précis.

L'isolation réseau est assuré par le composant **kube-proxy**