

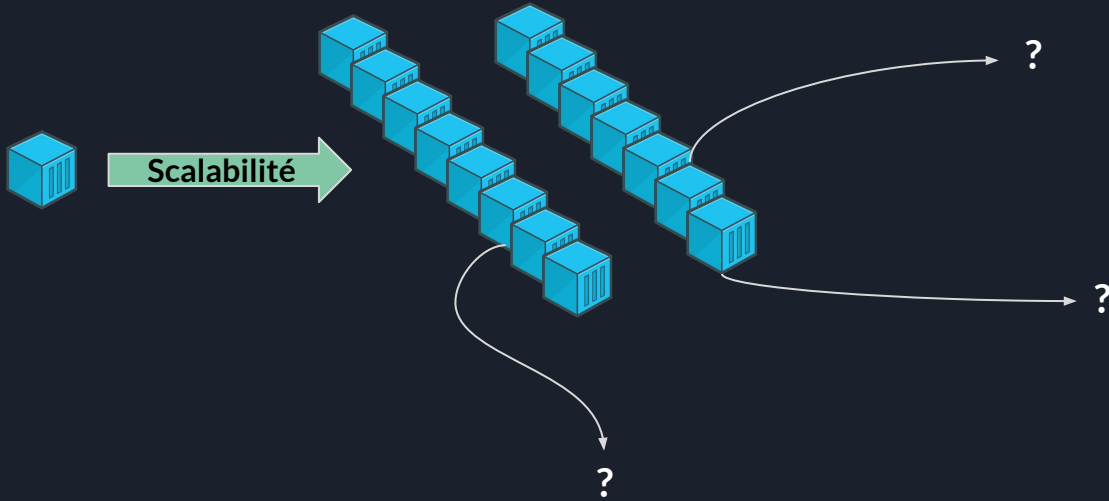


# Orchestration et Kubernetes

Un cours de Yann FORNIER

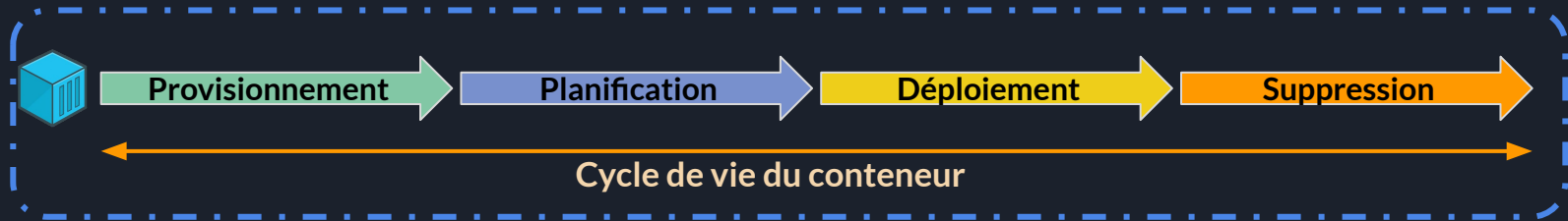
# Présentation de l'orchestration

L'orchestration de conteneurs est le processus d'**automatisation** de la **mise en réseau** et de la gestion de conteneurs pour déployer des applications à grande échelle. Les architectures de microservices peuvent comporter des **centaines** voire des **milliers** de conteneurs au fur et à mesure que les applications se développent.



# Automatisation du cycle de vie d'un conteneur

Les outils d'orchestration de conteneurs visent à simplifier la gestion des infrastructures de conteneurs en automatisant leur cycle de vie complet, depuis le provisionnement et la planification jusqu'au déploiement et à la suppression. Les entreprises peuvent bénéficier de la conteneurisation à grande échelle sans avoir à encourir de frais de maintenance supplémentaires.



# Pourquoi l'orchestration des conteneurs est nécessaire ?

Les conteneurs sont devenus l'unité informatique standard pour les applications cloud natives. Les fournisseurs de cloud proposent des instances de serveurs virtuels pour exécuter toutes sortes de charges de travail informatiques et conviennent parfaitement aux charges de travail basées sur des conteneurs. La seule exigence pour pouvoir exécuter des conteneurs est que le serveur lui-même exécute un service de conteneurisation tel que Docker.



*Le conteneur*  
*Unité informatique standard*

# Cas d'utilisation de l'orchestration de conteneurs

## Orchestration

Gestion et dimensionnement des conteneurs sur un certain nombre d'instances

Exécuter de nombreuses applications conteneurisées différentes

Exécuter simultanément différentes versions d'applications (test, prod, CI/CD)

Garantir la continuité du service des applications

Exécuter plusieurs instances d'une application dans plusieurs régions géographiques différentes

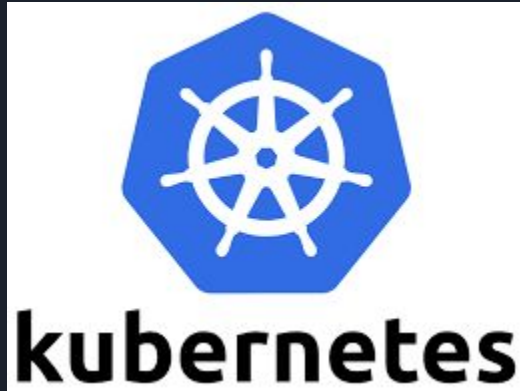
Optimiser l'utilisation de plusieurs instances de serveurs afin de réduire le coût global

Exécuter des applications complexes contenant des centaines de microservices différents

# Introduction à Kubernetes

Kubernetes (ou K8s) est un système open-source permettant l'automatisation du déploiement, la gestion et la scalabilité des applications conteneurisées. L'outil a été développé initialement par Google et est maintenant maintenu par la CNCF (Cloud Native Computing Foundation).

Kubernetes offre une approche déclarative pour gérer les infrastructures de conteneurs, rendant les systèmes plus robustes et flexibles à maintenir.





# Objets de Kubernetes

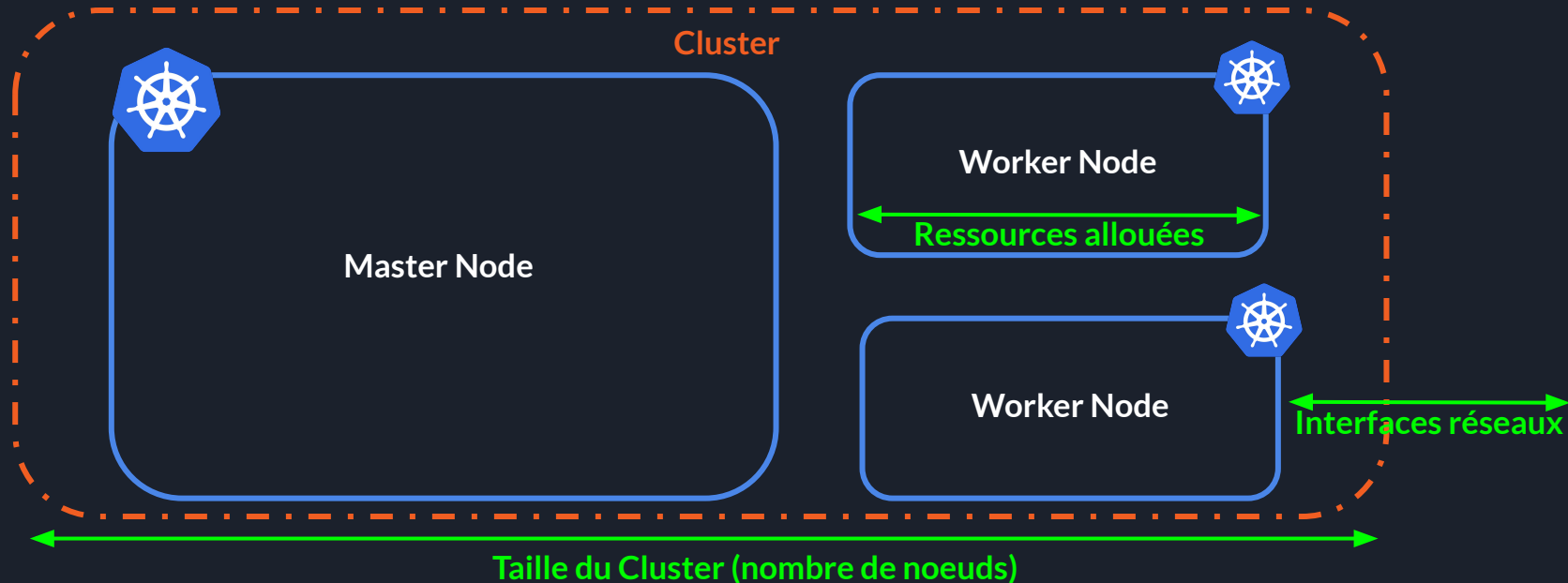
**Les noeuds**

**Les pods**

**Les services**

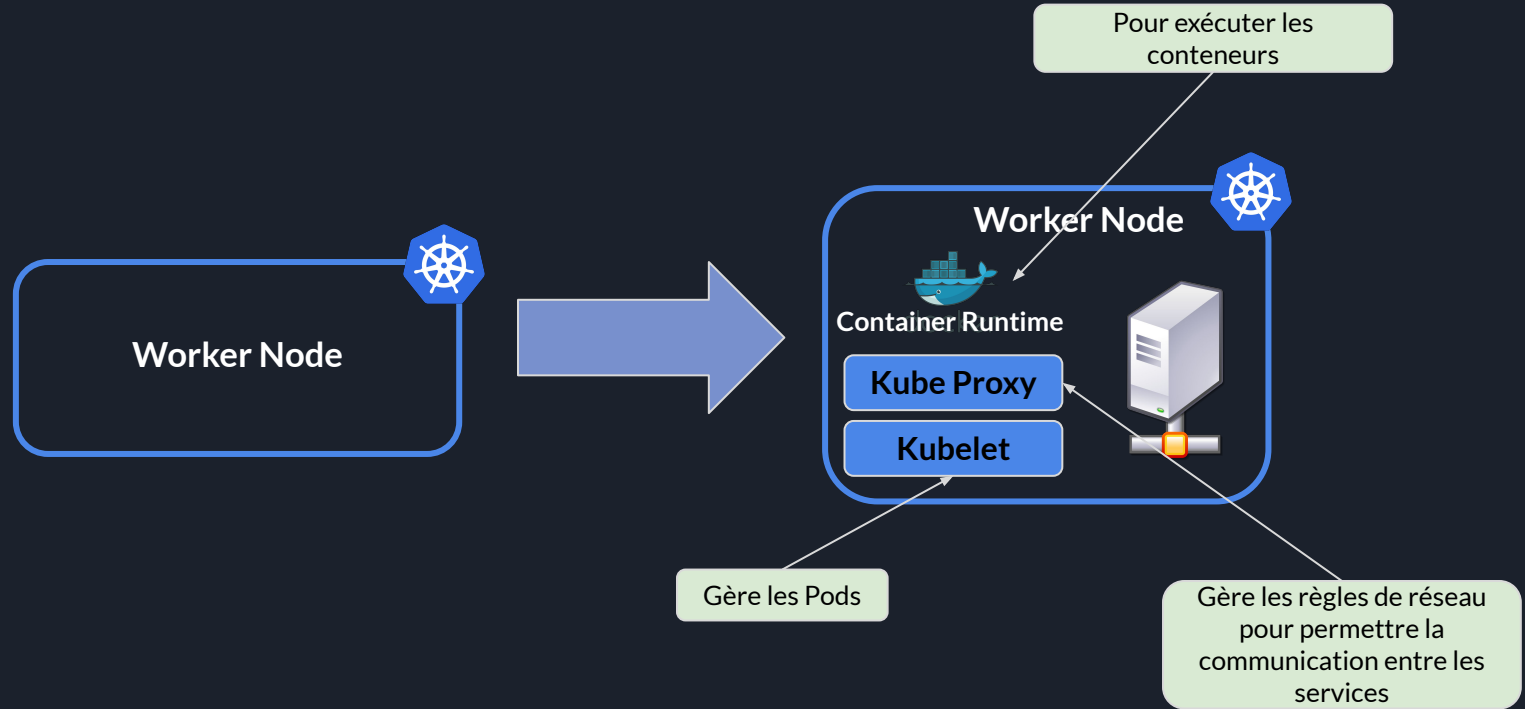
# Les Noeuds de Kubernetes

Un **Node** est un serveur physique ou une VM du Cluster Kubernetes qui exécute des conteneurs. Souvent, on ne gère pas les serveurs dans les Clouds publics, mais on doit gérer la **taille du cluster**, les **ressources allouées** à chaque noeud et les **interfaces réseaux**.

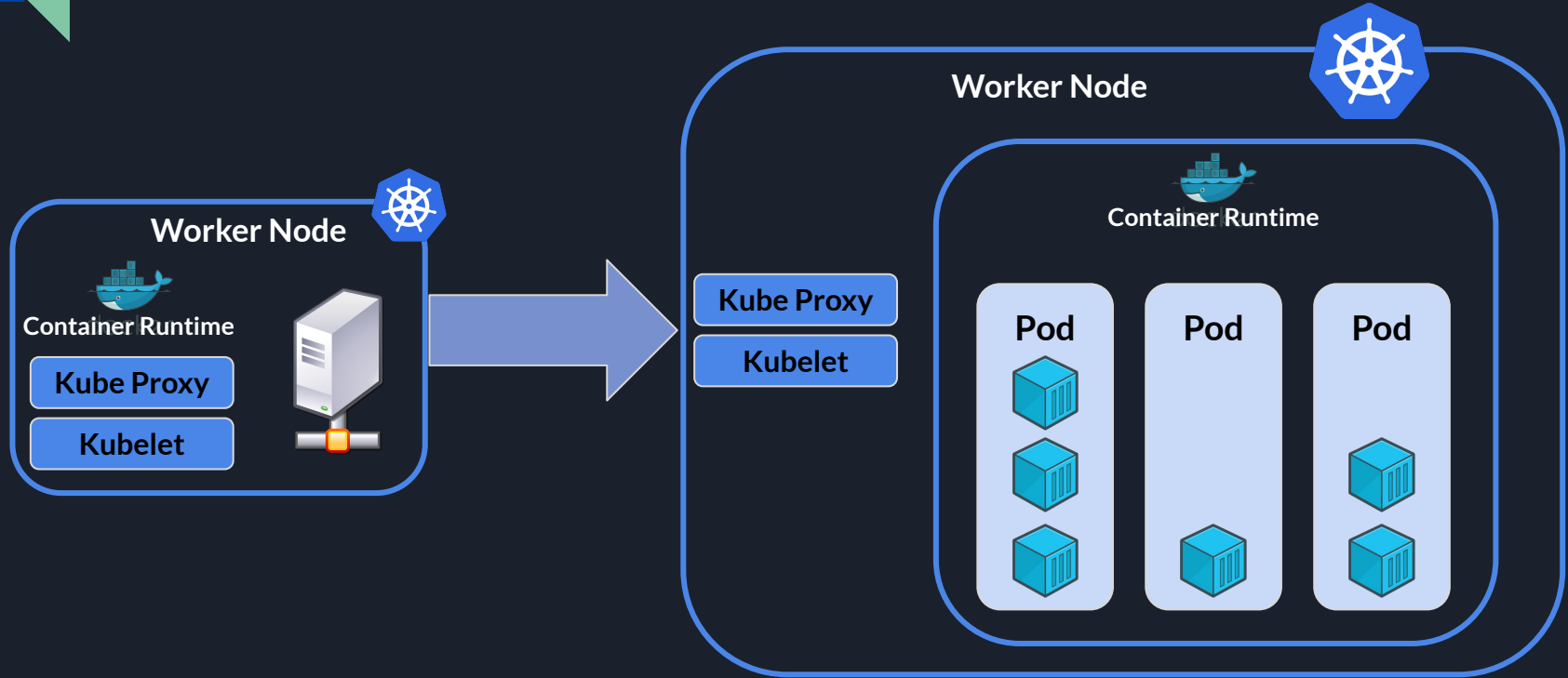




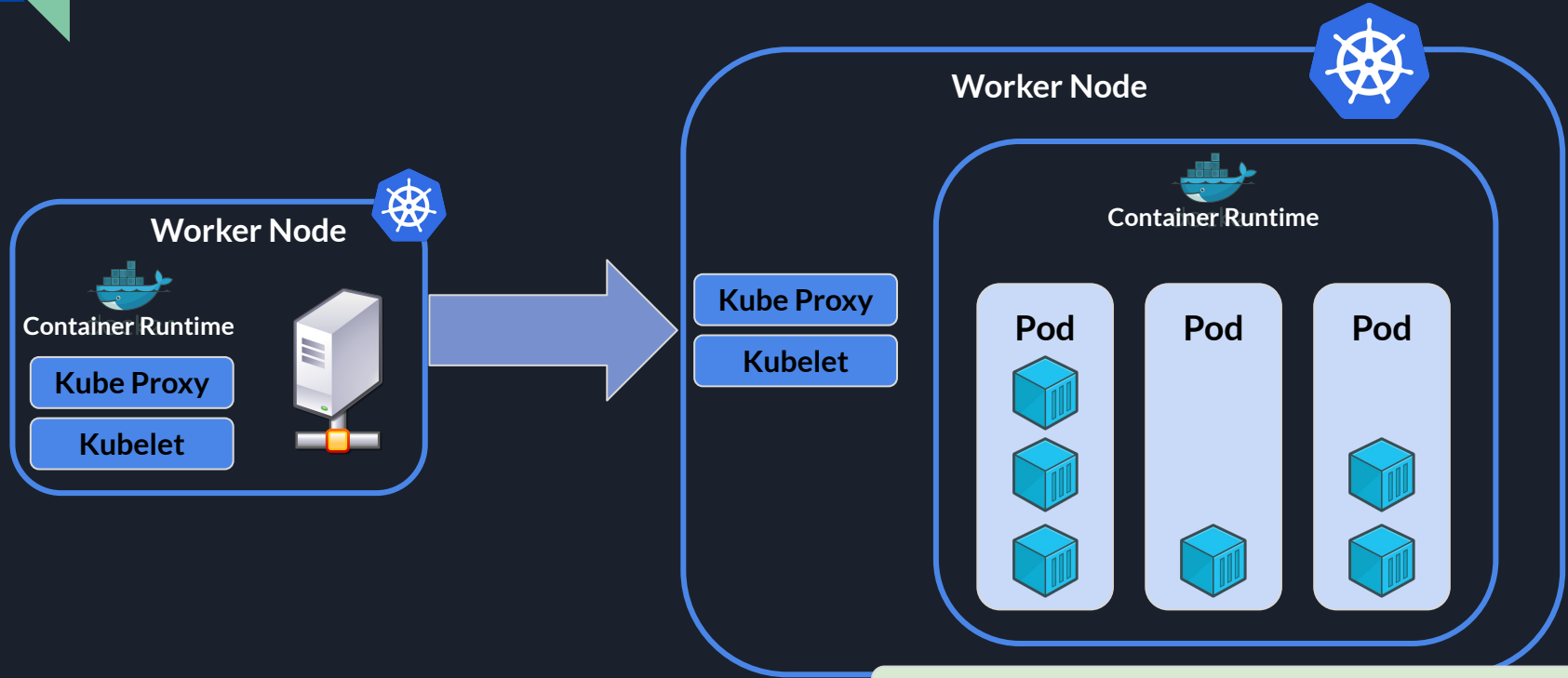
# Zoom sur le Worker Node



# Zoom sur le Worker Node



# Zoom sur le Worker Node

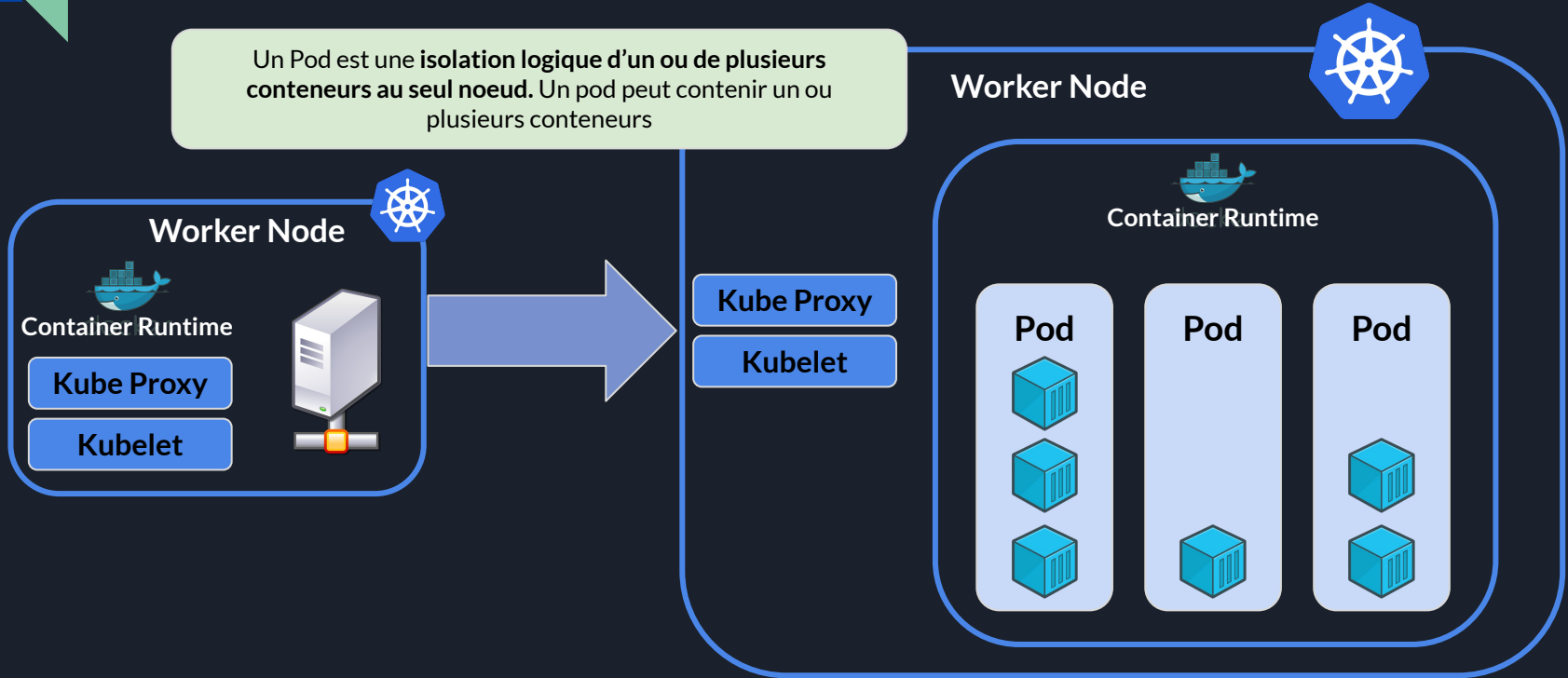


Les **Pods** sont les plus petites unités informatiques déployables qui peuvent être créées et gérées dans Kubernetes

NB : "Pods" : "Groupe de baleines"

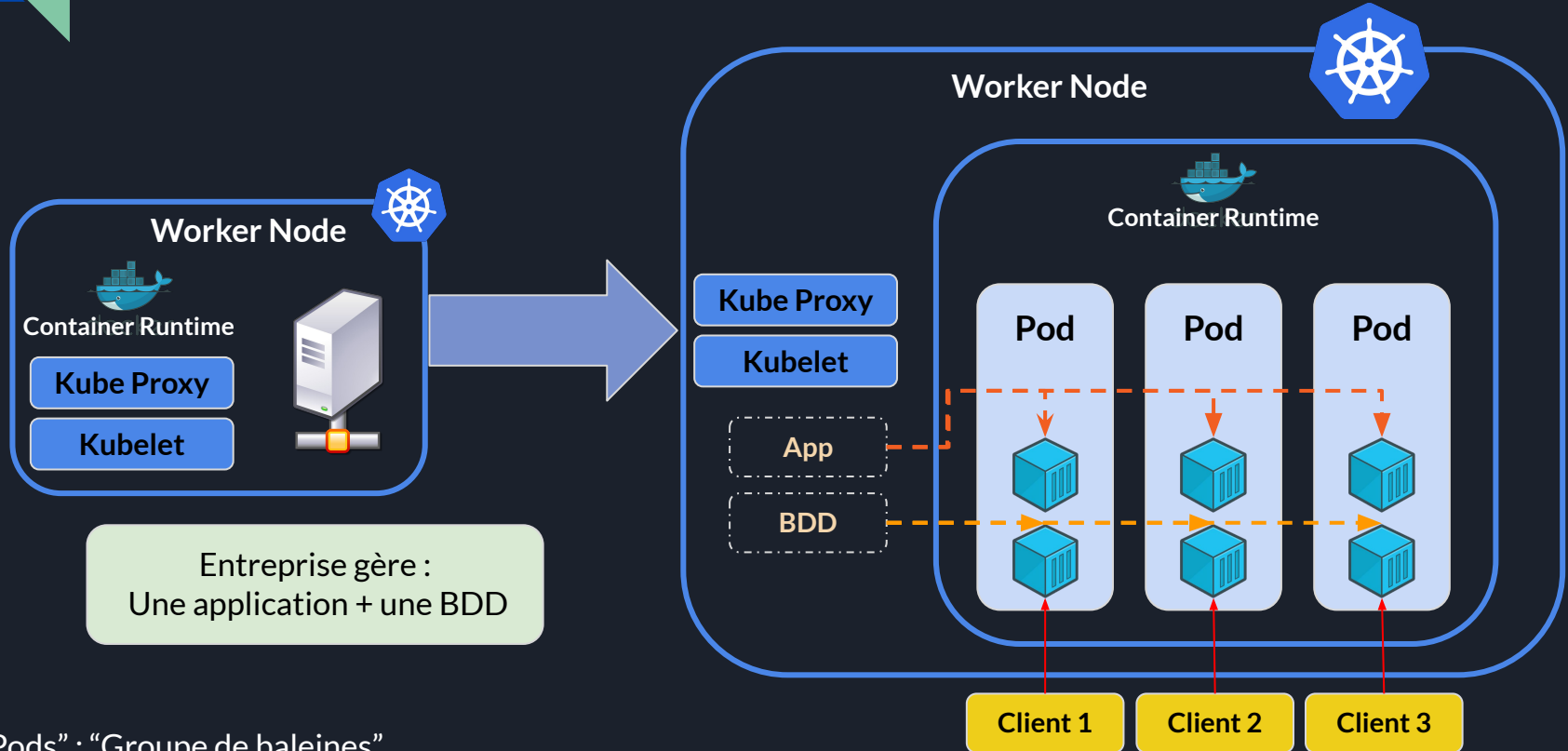
# Zoom sur le Worker Node

Un Pod est une **isolation logique d'un ou de plusieurs conteneurs au seul noeud**. Un pod peut contenir un ou plusieurs conteneurs



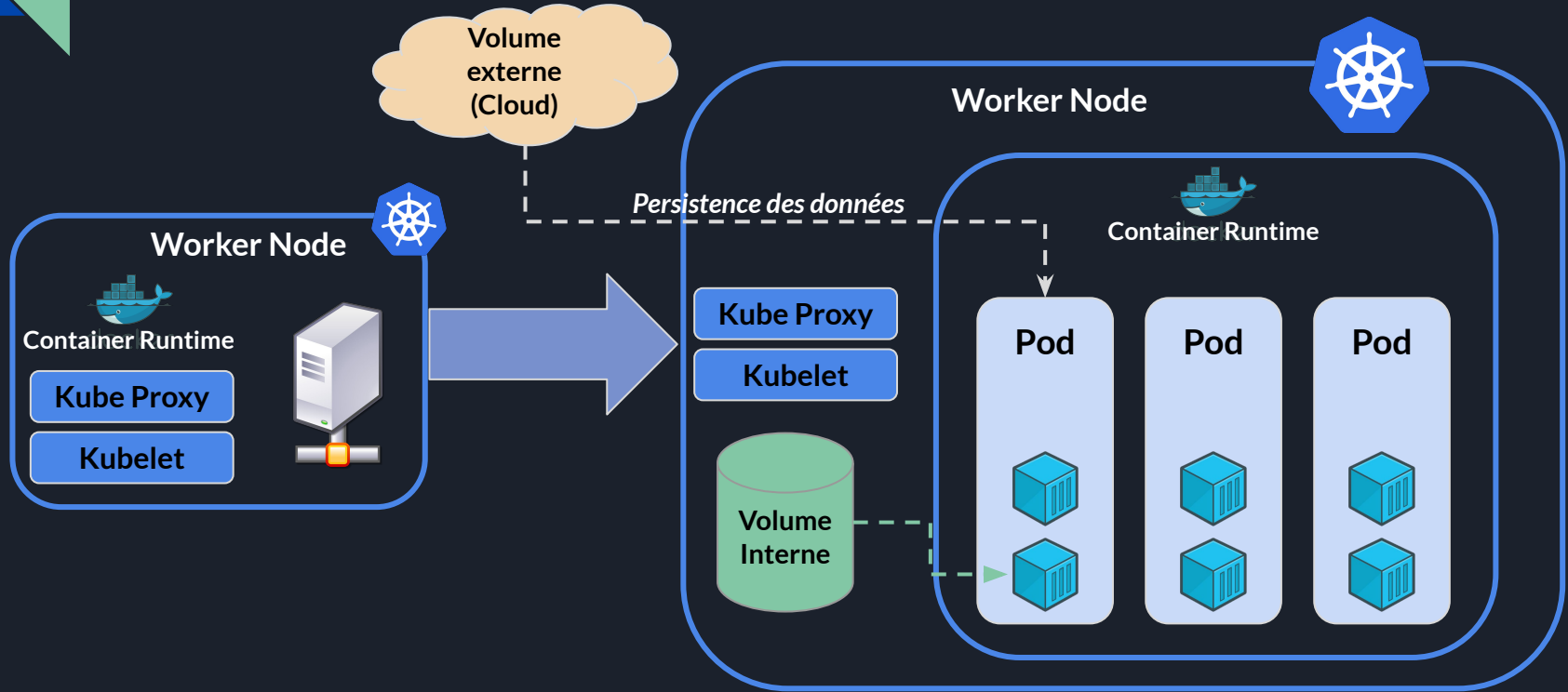
NB : “Pods” : “Groupe de baleines”

# Zoom sur le Worker Node



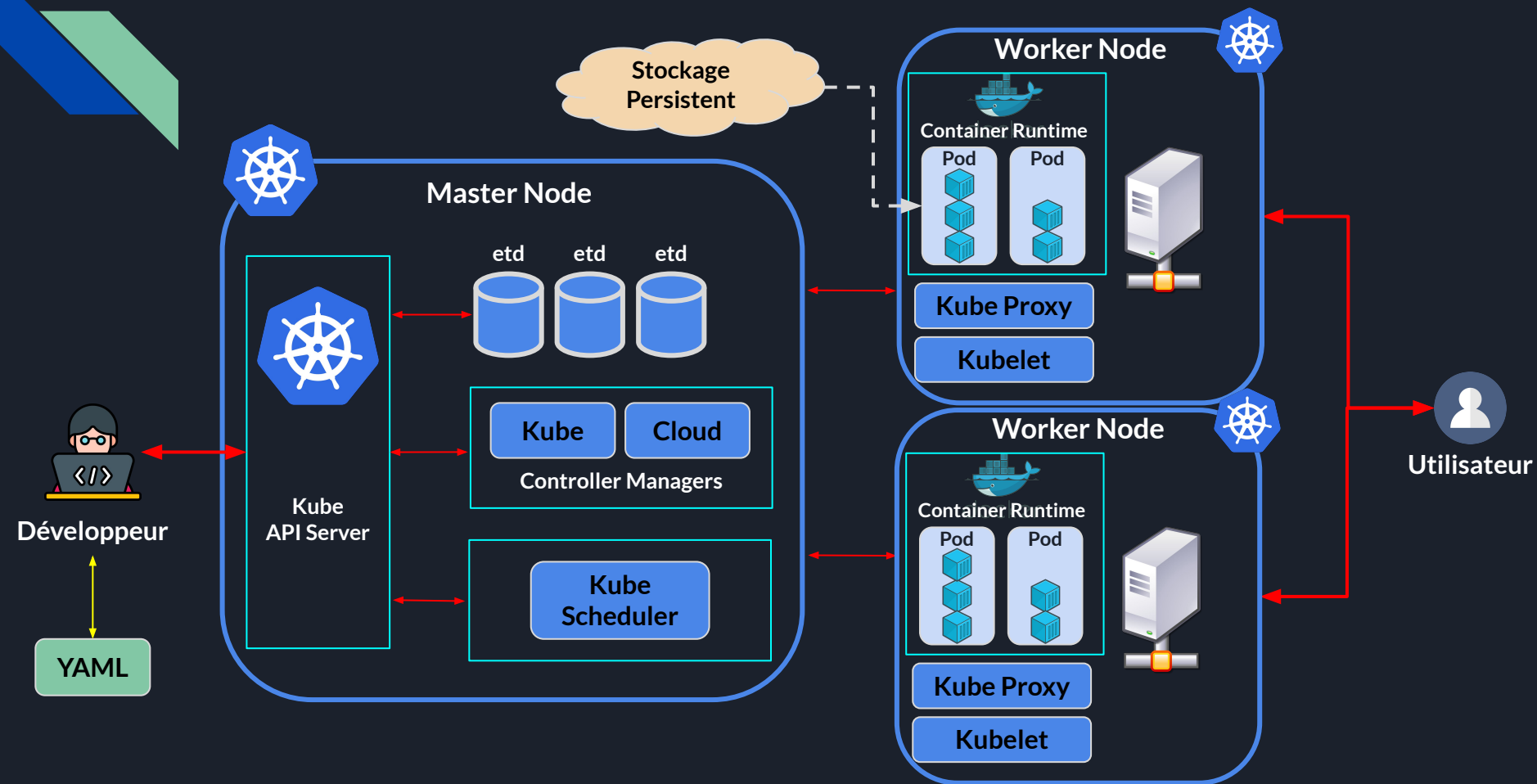
NB : "Pods" : "Groupe de baleines"

# Zoom sur le Worker Node

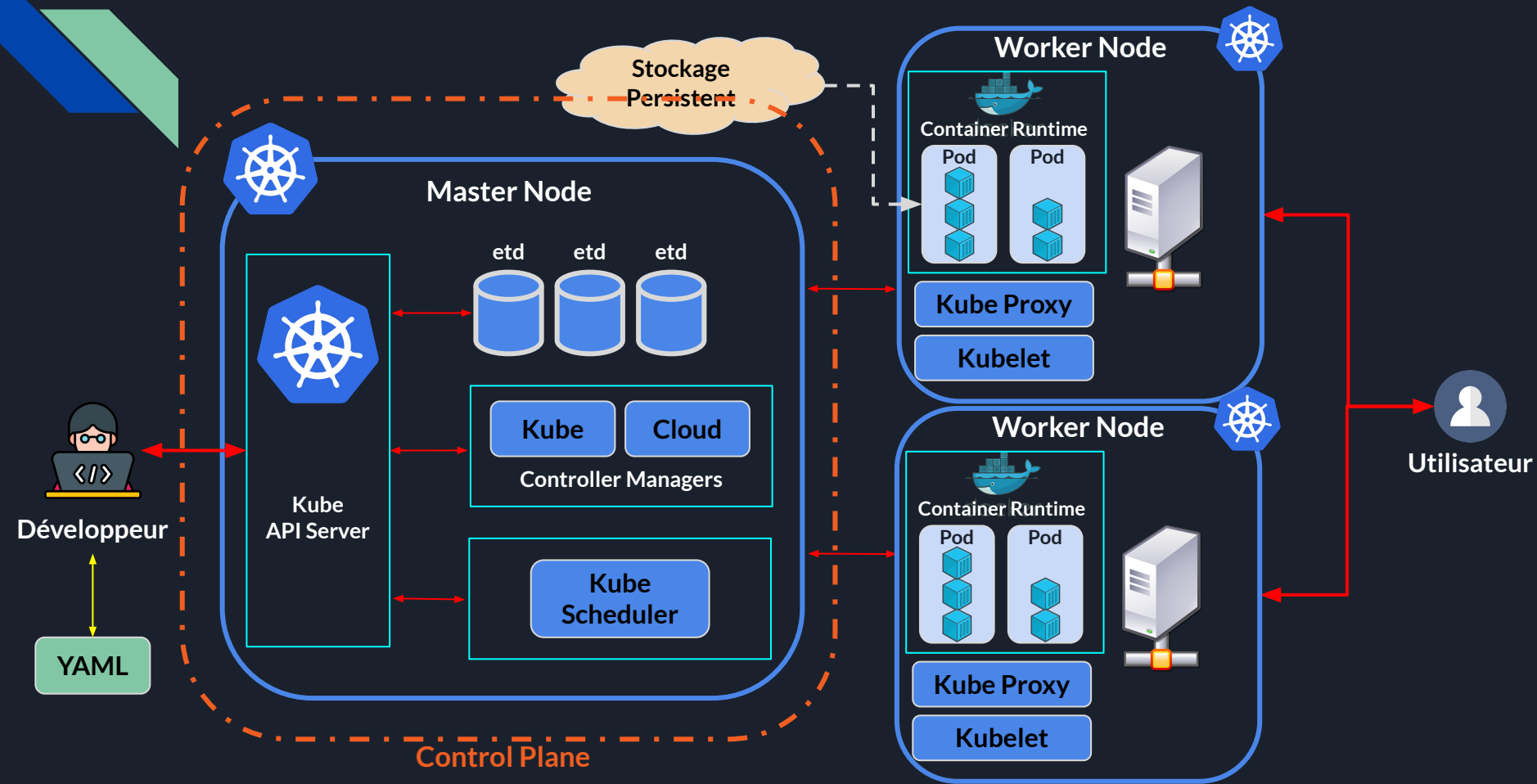


NB : "Pods" : "Groupe de baleines"

# Architecture de Kubernetes

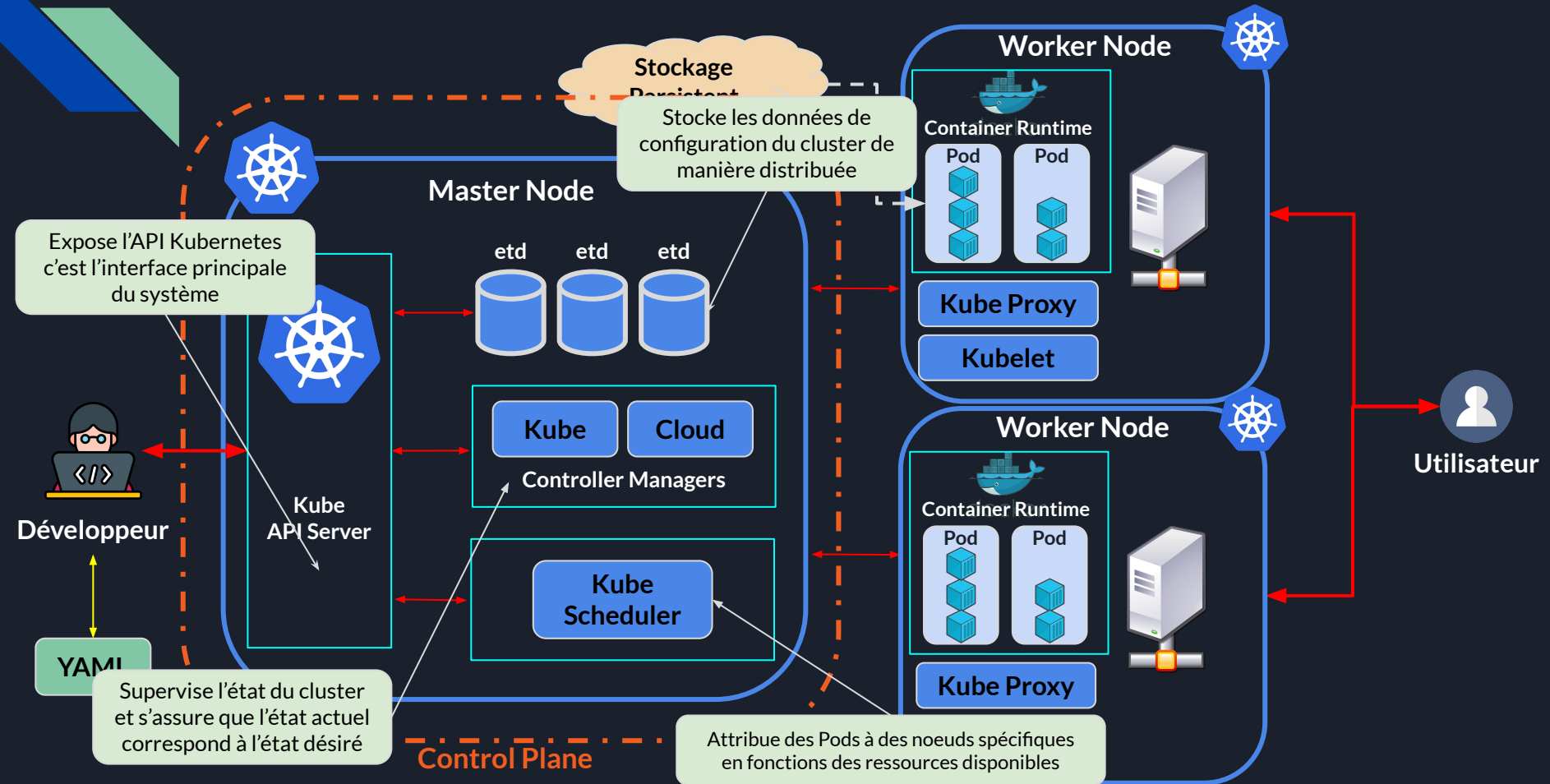


# Architecture de Kubernetes





# Architecture de Kubernetes



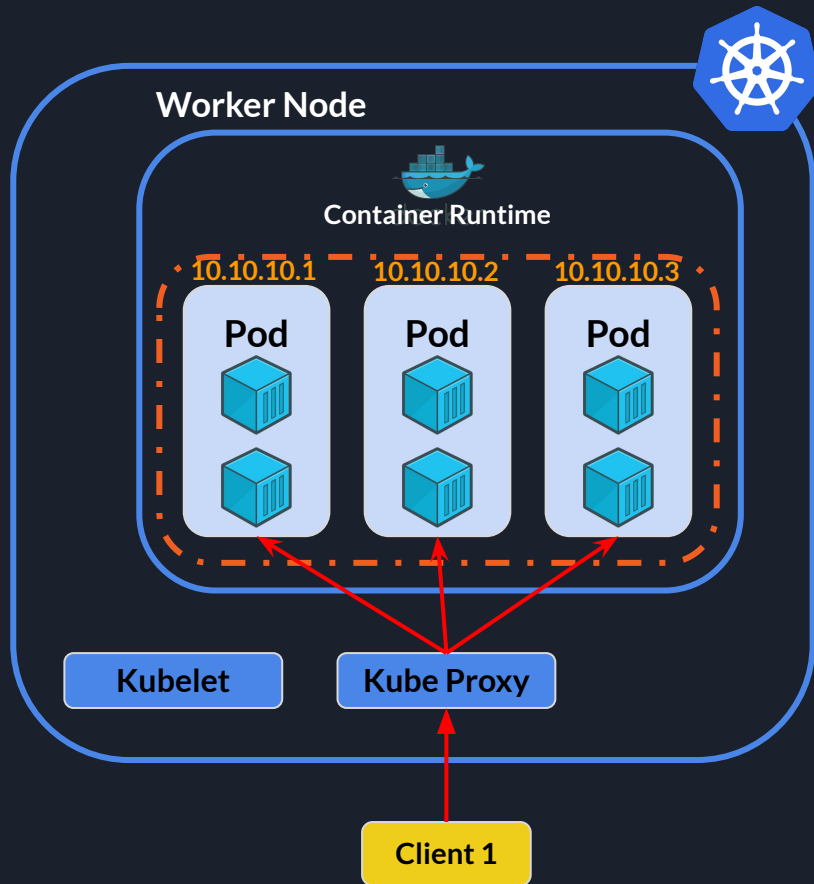


# Les services sur Kubernetes

Les services fournissent une couche d'abstraction qui englobe un ensemble de pods et permet un trafic extérieur par des services d'équilibrage de charge et de découverte. Dans un cluster K8s, chaque service représente potentiellement une application accessible depuis l'extérieur et exécutée dans plusieurs pods sur un ou plusieurs noeuds. Une adresse IP est attachée à ce service, permettant d'équilibrer le trafic parmi tous les pods du service.

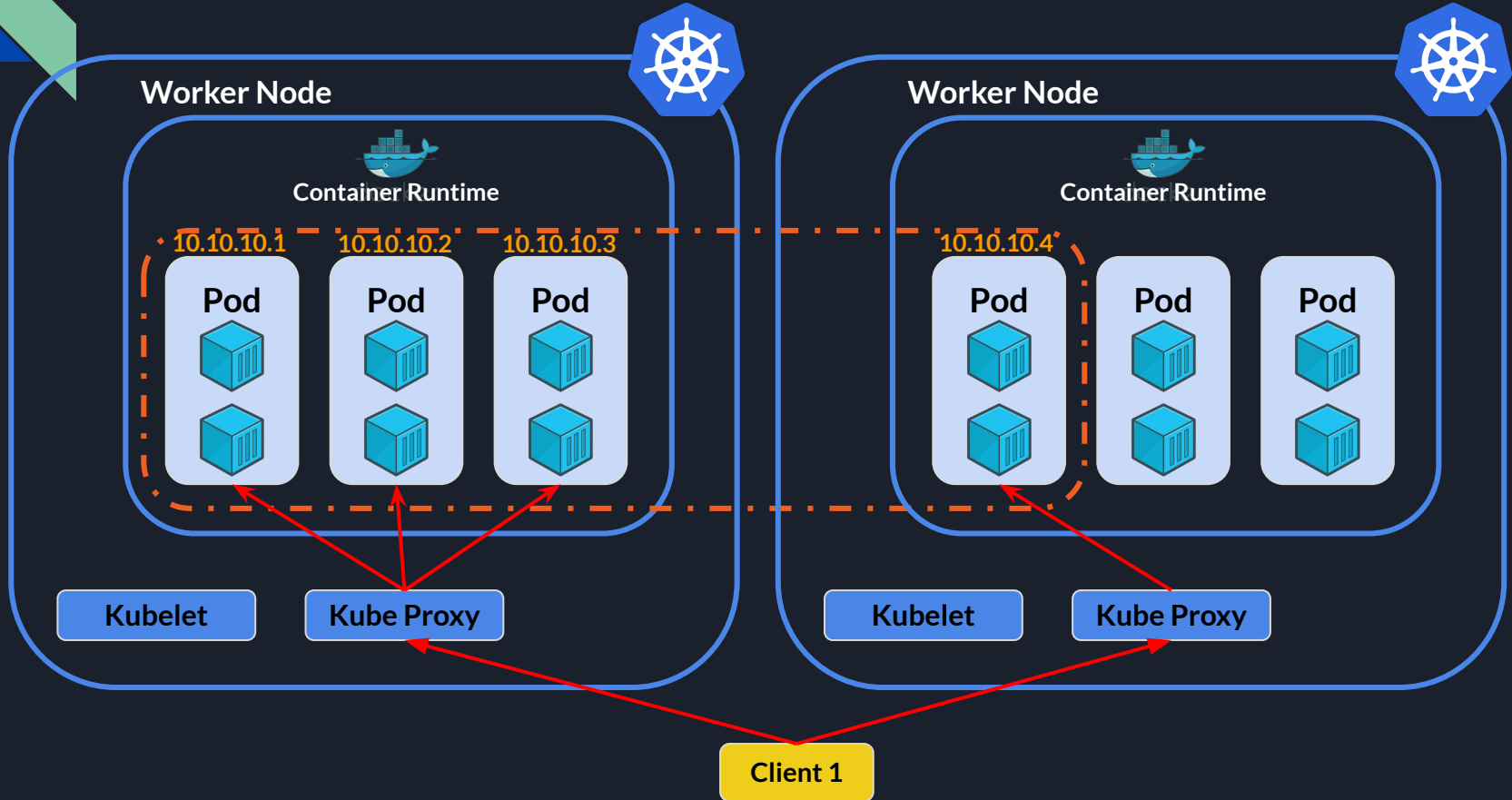
# Les services de Kubernetes

Entreprise gère :  
Une application + une BDD



# Les services de Kubernetes

Entreprise gère :  
Une application + une BDD





# TP : Installation et configuration d'un Cluster Kubernetes en local via Minikube

Téléchargez et réalisez le TP noté "Installation et Configuration d'un Cluster Kubernetes sur Windows" présent sur le GitHub de la classe :

[https://github.com/Le-Minh-Phuc/MScDE/blob/main/TP Installation Configuration Kubernetes Windows.docx.pdf](https://github.com/Le-Minh-Phuc/MScDE/blob/main/TP%20Installation%20Configuration%20Kubernetes%20Windows.docx.pdf)

Rendu : Rapport de TP au format PDF contenant des captures d'écran et des textes d'explications concernant l'installation de Kubernetes.

Deadline : Dimanche 29 septembre 18h00

Adresse de rendu : yann.fornier@gmail.com

# Projet Docker+K8s à réaliser (en groupe de 3 personnes ou solo au choix)

## Projet

TP noté Docker

TP Noté K8s

Déployer une **BDD (PostgreSQL ou MySQL)** comme **Pod**  
Déployer une **application web** (serveur **NodeJS** ou **Django**)  
Créer un **service** pour **connecter** l'application web à la BDD  
Exposer l'application web à l'**extérieur** du cluster via un service de type **NodePort** ou **LoadBalancer**

Date de présentation opérationnelle : Mardi 26 novembre 2024 : 16h15 - 18h15

Deadline du rendu du rapport de projet : Dimanche 24 novembre 2024

Format : PDF comprenant l'architecture déployée, la présentation des outils, exemples de code utilisés

Adresse de rendu : [yann.fornier@gmail.com](mailto:yann.fornier@gmail.com)