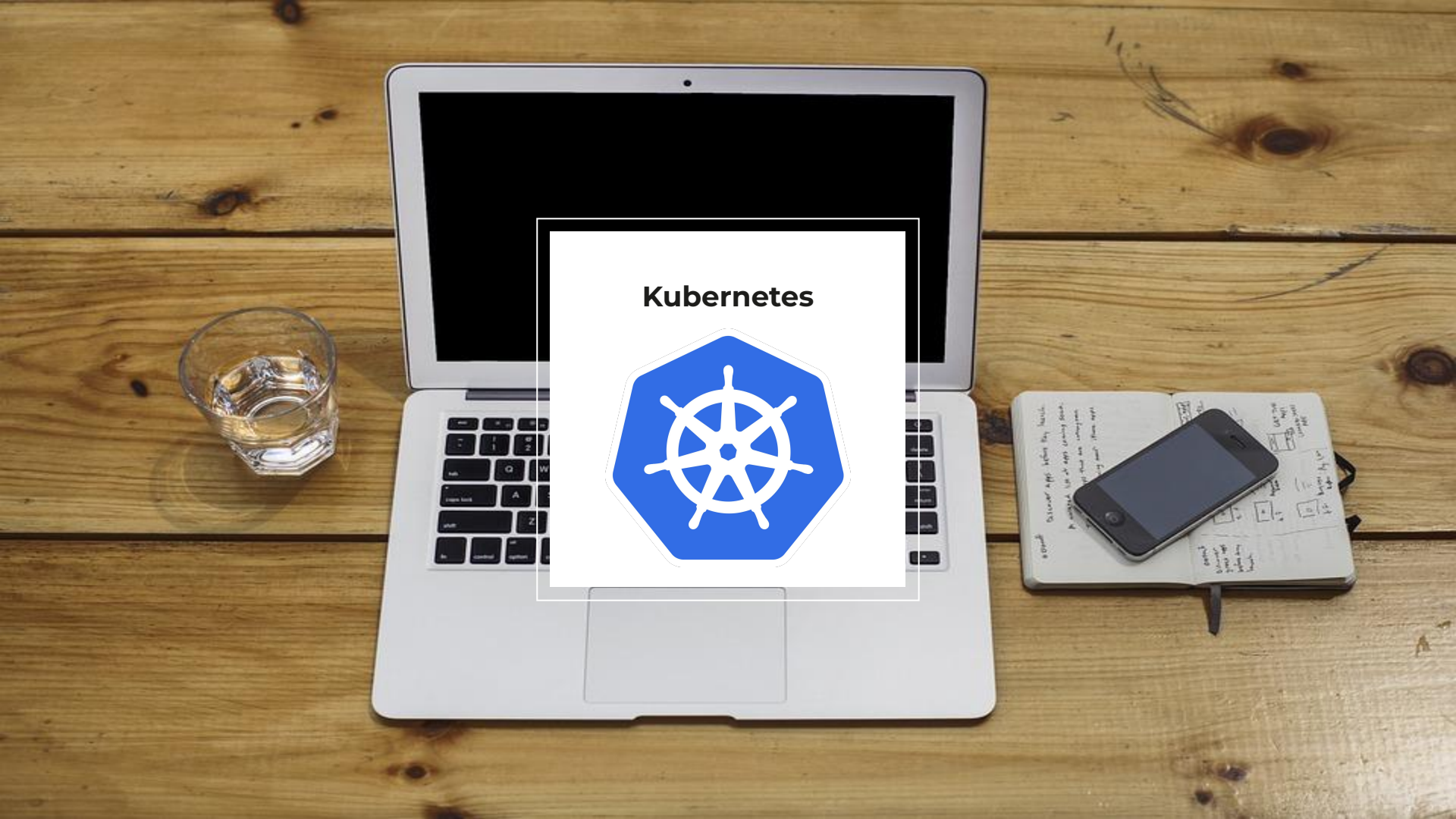
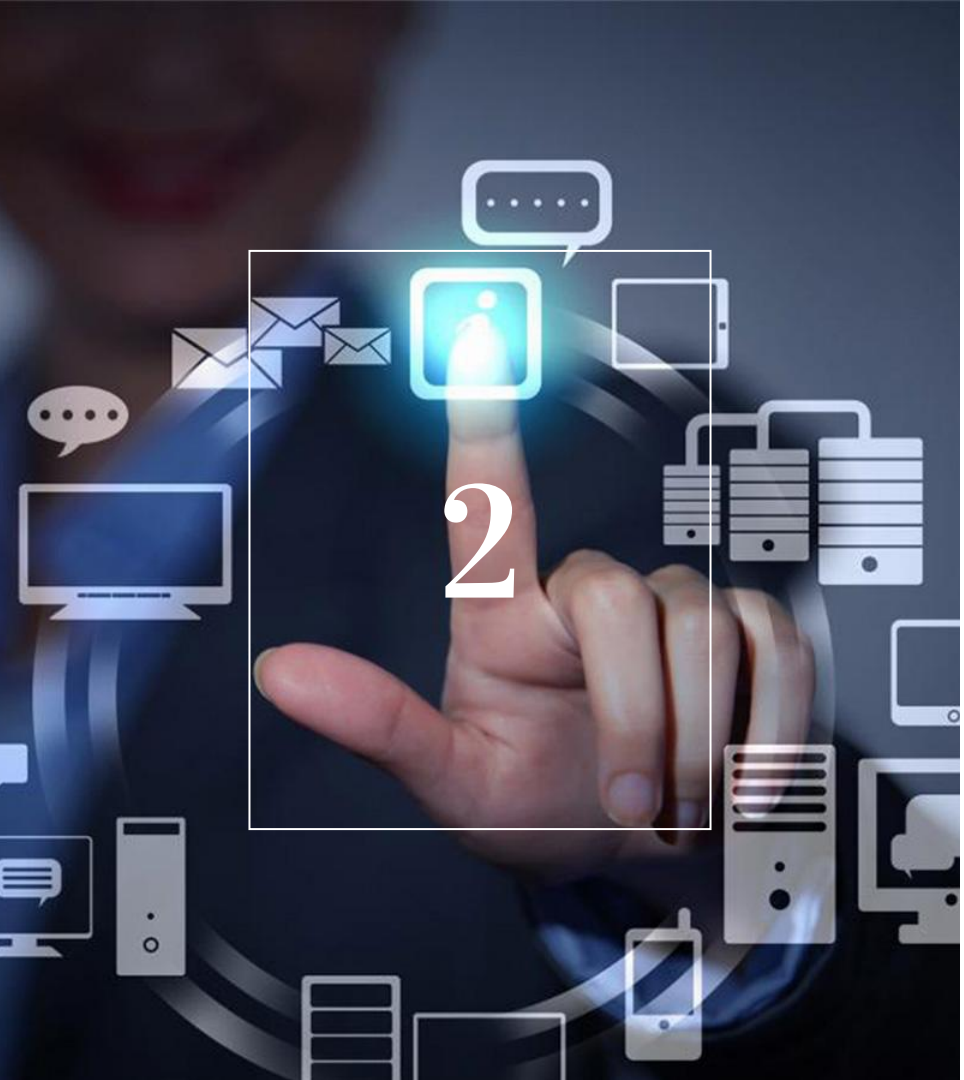

Kubernetes

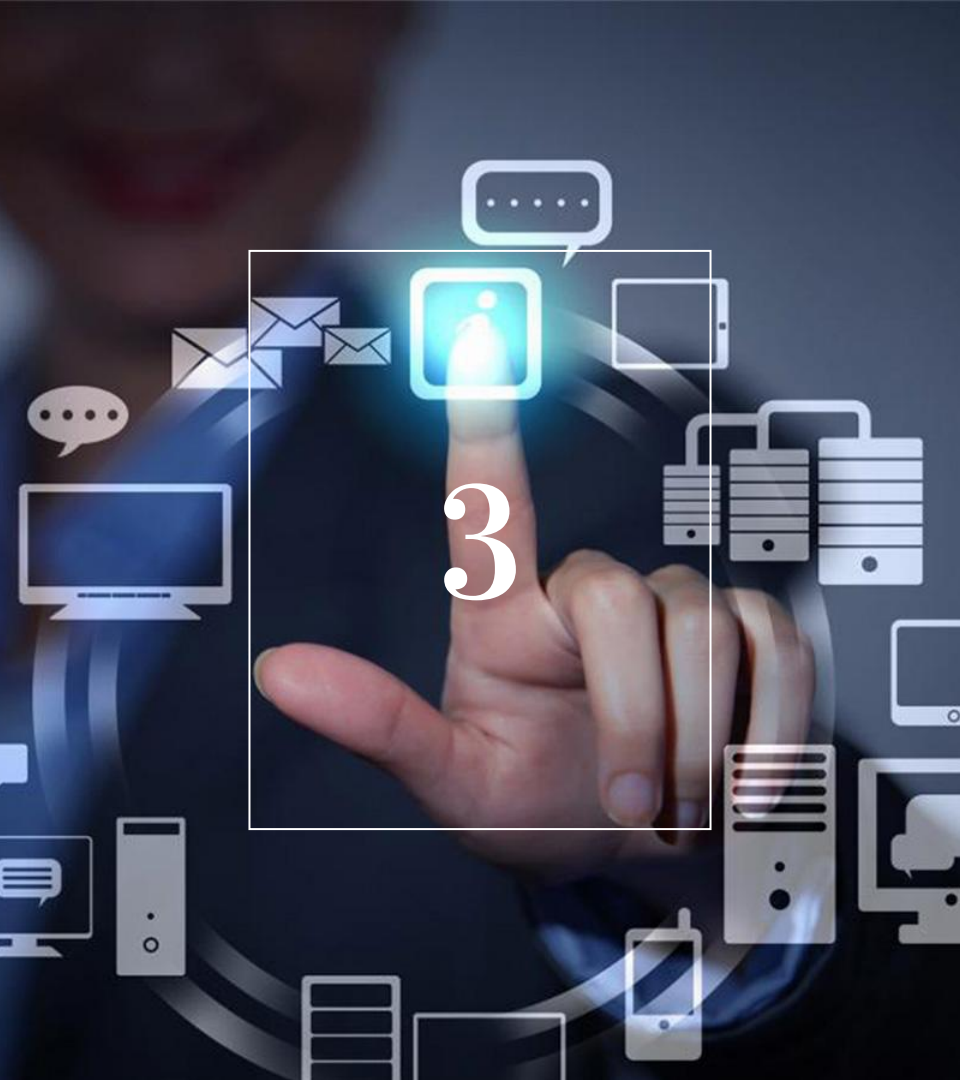


TP

- Créer un container contenant un projet Web avec un nginx et le serveur applicatif

- Créer deux containers 1 contenant nginx et 1 contenant le serveur applicatif

- Créer deux applications qui communiquent directement (micro-service) contenant l’architecture précédente



Historique et Idéologie

CREATION

Google a créé Kubernetes en 2015 (2014)

IDEOLOGIE

Open source : Plateforme gratuite et collaborative pour l’orchestration de conteneurs.

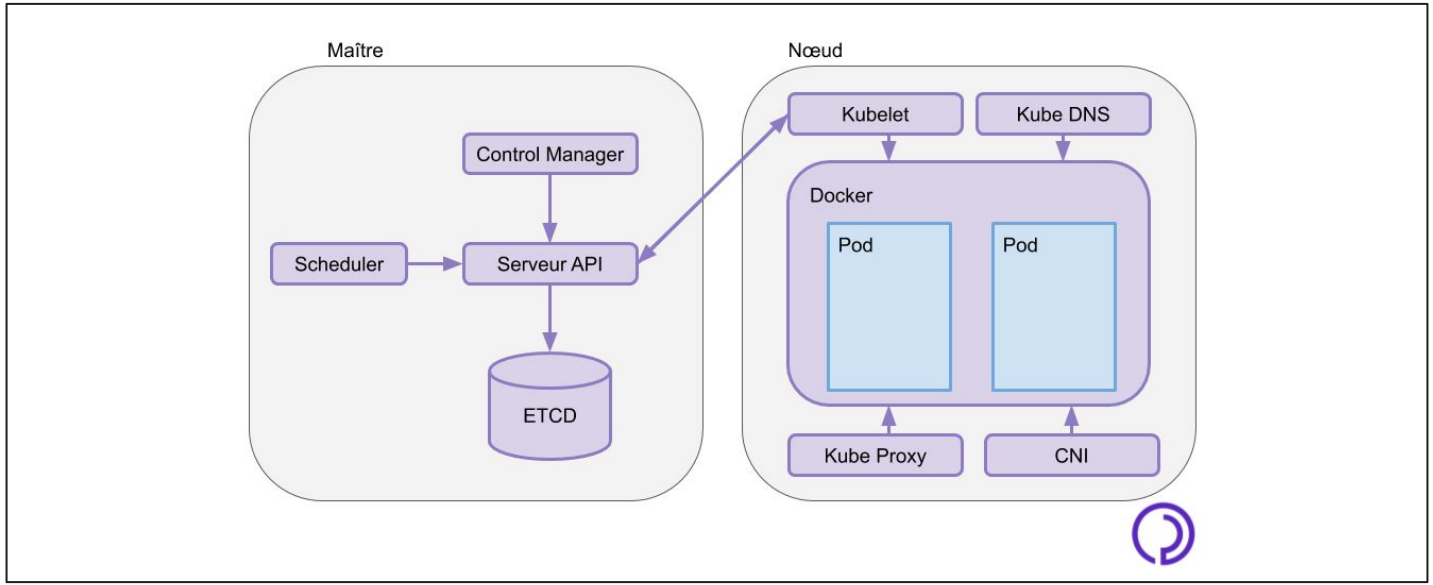
Automatisation : Facilite le déploiement, la mise à l’ échelle et la gestion des applications.

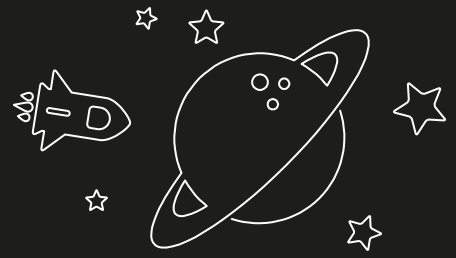
Gestion de clusters : Permet de coordonner plusieurs serveurs pour une haute disponibilité.

Microservices : Idéal pour déployer et gérer des architectures basées sur des microservices.

Interopérabilité : Compatible avec divers environnements cloud et sur site.

Infrastructure Kubernetes





Master

5

6

API Server

Interface principale : Il expose l’API REST de Kubernetes, permettant aux utilisateurs, aux outils et aux composants internes de communiquer avec le cluster.

Sécurité et contrôle : Il intègre des mécanismes d’authentification et d’autorisation pour garantir que seules les entités autorisées peuvent modifier l’état du cluster.

Coordination des composants : Il assure la communication entre les différents éléments du plan de contrôle (scheduler, controller-manager, etc.) et les nœuds de travail, facilitant ainsi la coordination et la gestion des applications.

7

ETCD

Stockage NoSQL : etcd est une base de données clé-valeur distribuée utilisée pour stocker l'état et la configuration du cluster Kubernetes.

Source de vérité du cluster : Toutes les informations critiques sur l'état du cluster (par exemple, la configuration des ressources) sont centralisées dans etcd, ce qui en fait la source de vérité pour Kubernetes.

Haute disponibilité : Sa conception distribuée permet d'assurer une continuité de service même en cas de défaillance partielle du système.

8

Scheduler

Évaluation des contraintes : Il prend en compte divers critères tels que l’affinité, la tolérance aux pannes, les ressources requises, et les politiques de qualité de service pour optimiser le placement.

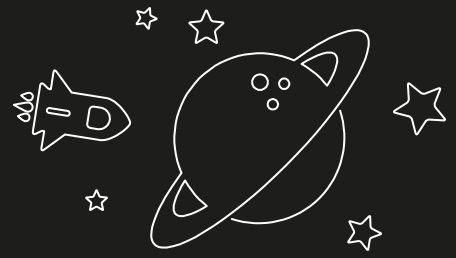
Optimisation de l'utilisation des ressources : Il s’appuie sur le control manager, pour déployer les pods sur les noeuds les moins occupés

9

Control Manager

Surveillance et réconciliation : Le control manager surveille l’état du cluster via l’API et déclenche des actions correctives

Automatisation des tâches : En automatisant des opérations telles que le redémarrage de pods défaillants ou l’ajustement des réplicas,



Noeud

10

11

Kubelet

Agent de nœud

Gestion des conteneurs : Il interagit avec le runtime de conteneurs

Surveillance de la santé : Il réalise des contrôles réguliers de l'état des pods et des conteneurs

Monitoring : Reporte ces informations au serveur API

12

Container runtime

Deux possibilités principales :

- Containerd
- Docker

Gestion des images : Il se charge du téléchargement, du stockage et de la gestion des images de conteneurs conformément aux spécifications OCI.

Cycle de vie des conteneurs : Il orchestre le démarrage, l'arrêt et la surveillance des conteneurs pour assurer leur bon fonctionnement.

13

KubeDNS / CoreDNS

Service de résolution DNS : KubeDNS assure la résolution des noms de services et de pods au sein du cluster Kubernetes, permettant aux applications de communiquer sans utiliser d’adresses IP.

Communication simplifiée : En attribuant des noms DNS aux services, il facilite la découverte des ressources et améliore la communication interservices.

CoreDNS : Nouveau service par défaut

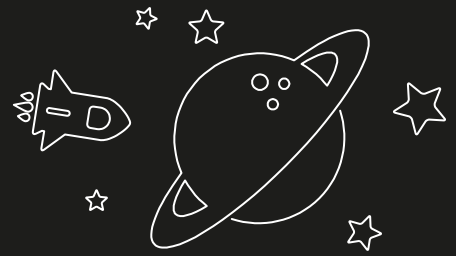
14

KubeProxy/CNI

Agent réseau : KubeProxy s'exécute sur chaque nœud du cluster et gère le routage du trafic réseau vers les services/pods.

Routage et load balancing : Il utilise des règles basées sur iptables ou ipvs pour rediriger efficacement le trafic vers les pods correspondants aux services.

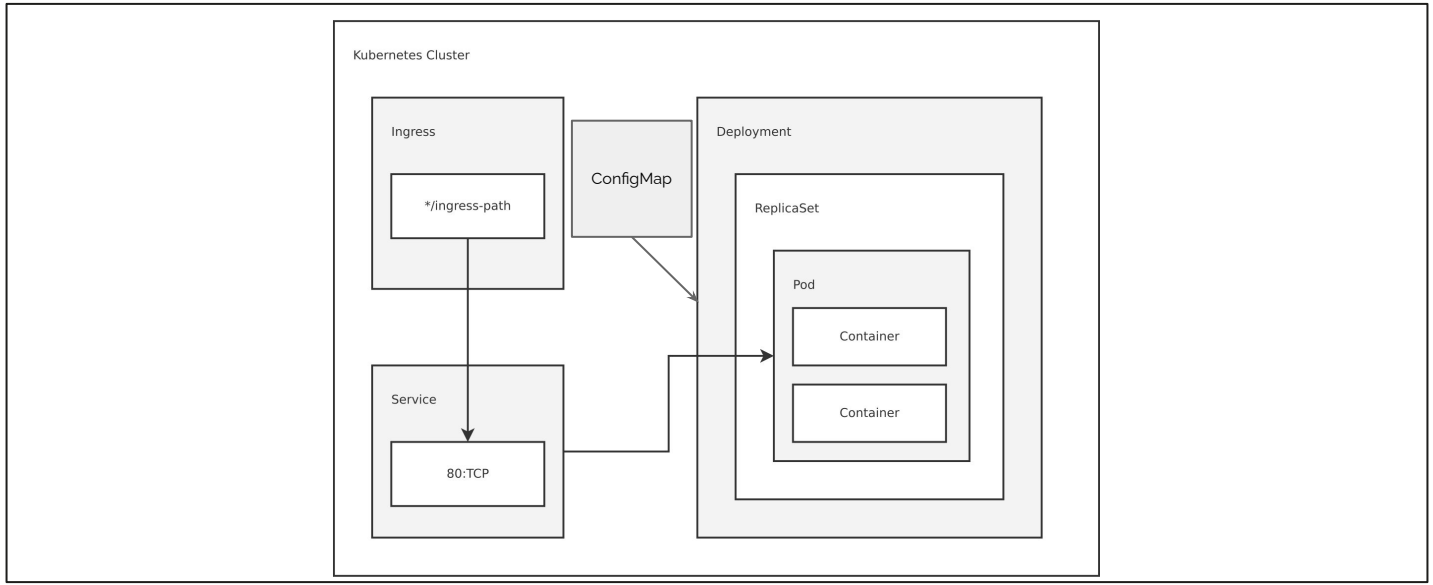
Mise à jour dynamique : KubeProxy surveille l'API Kubernetes pour adapter en temps réel ses règles de routage lors des modifications du cluster.



Architecture logicielle

15

Infrastructure Déploiement



17

Bah c’est un container

18

Pod

Unité de base : Regroupe un ou plusieurs conteneurs.

Cloisonnement : Tous les conteneurs d’un même pod partagent un espace réseau et des volumes de stockage.

Éphémérité : Les pods sont censés être temporaires et peuvent être recréés ou remplacés par les contrôleurs (ReplicaSets, Deployments) afin de maintenir l'état désiré du cluster. Naissance des infrastructures serverless / cloud functions

19

Déploiement

Fichier de configuration de l’application

Gestion déclarative : Permet de définir l’état voulu de l’application démarre/stoppe/met à jour. La configuration est ensuite stocké dans l’**etcd** et le **control manager** va garantir le maintien de cette configuration

Scalabilité et ReplicaSet : Permet de définir les règles de scalabilité et de réplications

Mises à jour progressives : Permet de définir les règles de mise à jour qui peuvent permettre d’éviter que l’application soit incaccessible pendant le déploiement. Permet de revenir en arrière en cas de problème (récupération de la configuration précédente)

Format : YAML/JSON

20

Service

Se rapproche de la notion d’application

Adresse IP et DNS stables : Permet d’obtenir une IP interne et des noms DNS internes pour accéder aux pods

Utilisation des sélecteurs : Utilise un système de matching de labels pour associer des pods à un service.

Load balancing et scalabilité : Répartit le trafic réseau entre les pods correspondants, gère la création/suppression de pods si nécessaire

Types de services :

- *ClusterIP* : Accessible dans Kubernetes
- *NodePort* : Routing NAT
- *LoadBalancer* : LB externe

21

Ingress

Agent réseau

Routage : Il route les requêtes vers les services associés selon la configuration. Permet de rendre les services exposables en dehors de Kubernetes

Ingress Controller : Souvent NGINX (traefik, HAProxy). Il met en place les règles de routage spécifiés dans la configuration Ingress

Surveillance de la santé : Des règles de checks de santé (Lancé, Prêt) peuvent être mise en place