

02 - Mise en Place d'un Cluster Kubernetes

Ce chapitre couvre l'utilisation de l'outil léger et rapide k3sup pour déployer Kubernetes sur des environnements légers ou dans le cloud.



par Tancrede SUARD

Code : KUB-A-1 (2024)

Préparation de l'Environnement

Machines Virtuelles

Préparer l'environnement cible, que ce soit des machines locales, des machines virtuelles ou des serveurs en cloud.

Configuration Système

Vérifier que l'environnement est prêt pour le déploiement de Kubernetes.

Prérequis Techniques

S'assurer que tous les prérequis techniques sont en place avant de commencer le déploiement.

Choix de l'Infrastructure

Développement local

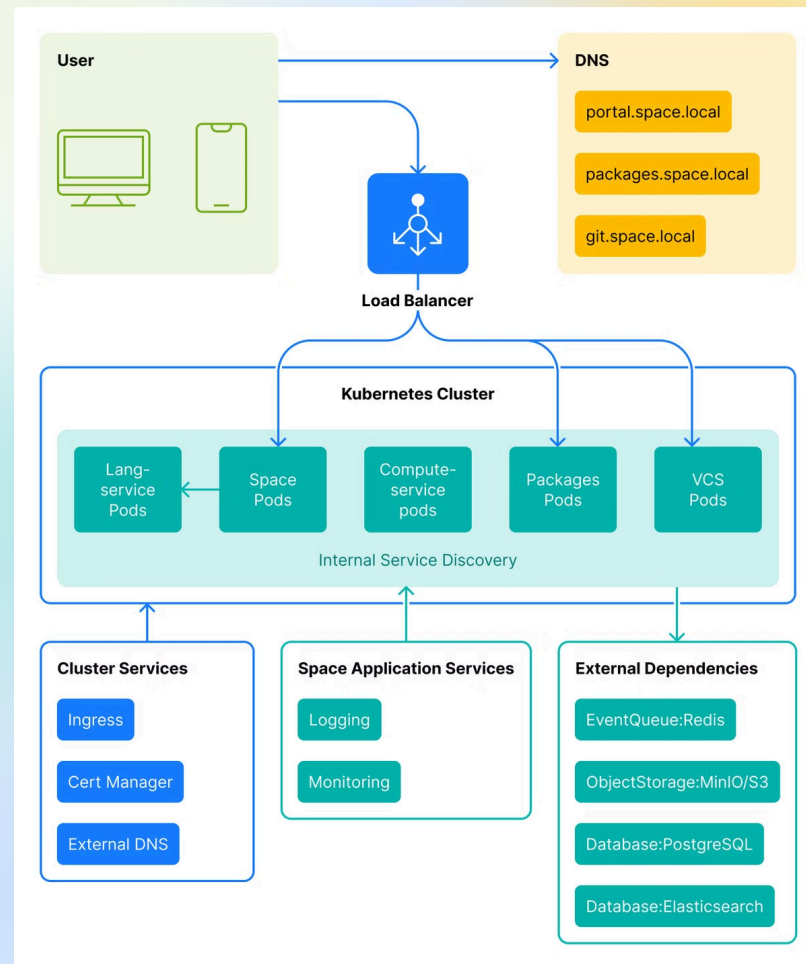
Machines virtuelles ou Raspberry Pi pour un environnement de test.

Environnements on-premise

Serveurs physiques au sein d'une entreprise.

Cloud

AWS, Azure, GCP ou autres fournisseurs cloud. K3sup facilite l'installation.



Prérequis Techniques



SSH Configuré

k3sup utilise SSH pour se connecter aux nœuds et y installer Kubernetes.



Accès Root

Permissions sudo requises sur les machines d'installation.



k3sup Installé

Installer k3sup sur la machine de contrôle pour l'installation de Kubernetes.

Installation de Kubernetes avec k3sup



Simplicité

Déploiement automatisé



Rapidité

Installation rapide

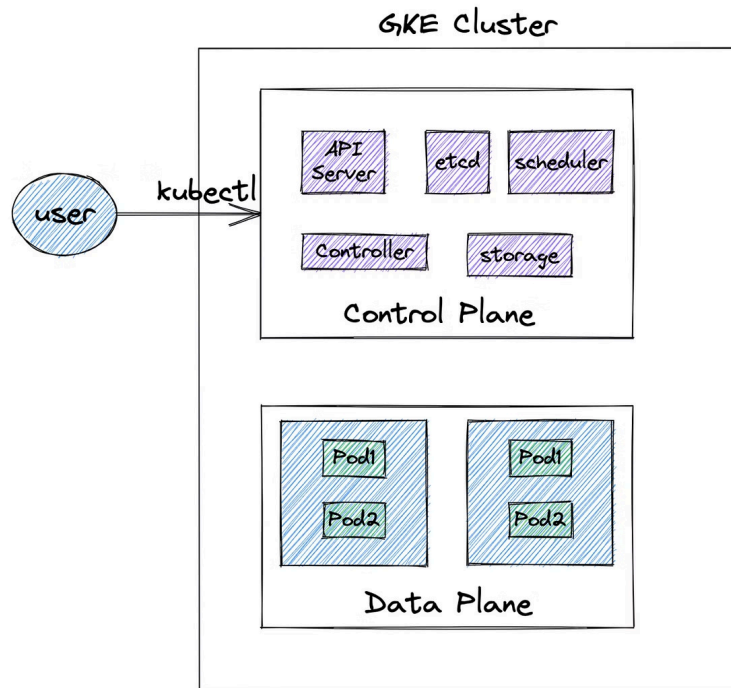


Légèreté

Utilisation de k3s

k3sup permet d'installer Kubernetes de manière simple et rapide en automatisant le déploiement de **k3s**, une version allégée de Kubernetes adaptée aux environnements à faible empreinte mémoire.

Déploiement du Nœud Maître



1

Exécuter la commande

Lancez la commande **k3sup install --ip <IP_MASTER> --user** depuis votre machine de contrôle.

2

Remplacer les paramètres

Remplacez **<IP_MASTER>** par l'adresse IP du nœud maître et **--user** par le nom de l'utilisateur ayant accès à cette machine.

3

Installer k3s

Cette commande installe k3s sur le nœud spécifié et configure **kubectl** pour se connecter au cluster.

Ajout de Nœuds de Travail

Exécuter la commande

```
k3sup join --ip --server-ip --user
```

Spécifier l'IP du nœud

Remplacez par l'adresse IP du nœud de travail que vous souhaitez ajouter au cluster.

Connecter au nœud maître

Remplacez par l'adresse IP du nœud maître et par le nom d'utilisateur ayant accès à la machine du nœud de travail.

Une fois le nœud maître configuré, vous pouvez ajouter des nœuds de travail pour répartir les charges de travail et améliorer les performances du cluster. Chaque nœud ajouté offre des ressources supplémentaires pour exécuter des pods et des conteneurs.

Vérification de l'Installation

1

Vérifier les Nœuds

Utilisez la commande **kubectl get nodes** pour afficher la liste de tous les nœuds du cluster et vérifier leur statut.

2

Connexion des Nœuds

Assurez-vous que tous les nœuds sont connectés et en bon état de fonctionnement.

3

Cluster Opérationnel

Vérifiez que le cluster Kubernetes est fonctionnel et prêt à accueillir vos déploiements.

Configuration et Gestion du Cluster

Configurer le Cluster

Définir les paramètres de configuration du cluster
Kubernetes pour garantir ses performances et sa fiabilité.

Maintenir le Cluster

Assurer un suivi régulier et des mises à jour pour garder le
cluster en bon état opérationnel.

Configurer kubectl

1

Vérification de la Configuration

Utilisez la commande **kubectl config view** pour afficher les informations de configuration de kubectl et les clusters configurés.

2

Configuration Automatique

Après l'installation, k3sup configure automatiquement kubectl pour qu'il pointe vers le cluster Kubernetes.

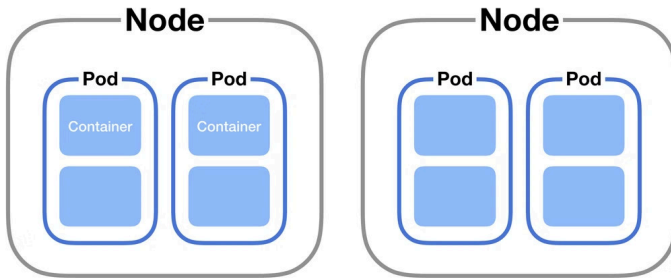
3

Interaction avec le Cluster

Grâce à cette configuration, vous pouvez désormais interagir avec le cluster Kubernetes à l'aide de kubectl.

Gestion des Ressources et Allocation

Cluster



Inspection des Nœuds

Utilisez **kubectl describe nodes** pour inspecter les ressources disponibles sur les nœuds.

Surveillance des Pods

Utilisez **kubectl get pods -A** pour vérifier les pods en cours d'exécution.

Allocation par le Scheduler

Le Scheduler Kubernetes attribue les pods aux nœuds en fonction des ressources disponibles.

Surveillance et Santé du Cluster

1 Vérifier l'État des Nœuds

Utilisez la commande **kubectl get nodes** pour surveiller l'état des nœuds du cluster.

2 Vérifier l'État des Pods

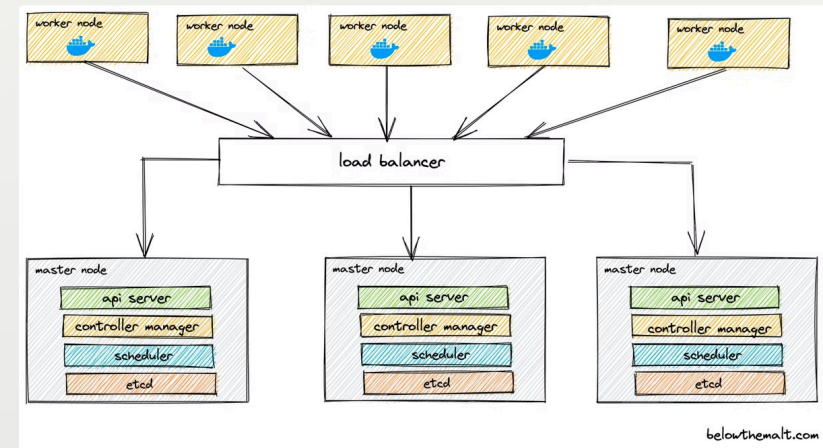
Utilisez la commande **kubectl get pods -n <namespace>** pour surveiller l'état des pods dans un namespace spécifique.

3 Outils de Surveillance Avancée

Intégrez des outils comme Prometheus pour obtenir des métriques détaillées sur la santé du cluster en production.

Maintien en Condition Opérationnelle (MCO) du Cluster

Le MCO est essentiel pour s'assurer que le cluster Kubernetes reste performant et sécurisé.



Surveillance du Cluster

1 Collecte de Métriques

Utiliser des solutions comme Prometheus pour surveiller les ressources (CPU, RAM, etc.).

2 Génération d'Alertes

Configurer des alertes en cas de défaillances pour une détection rapide des problèmes.

3 Visualisation en Temps Réel

Déployer Grafana pour visualiser l'utilisation des ressources du cluster.



Sécurisation du Réseau avec des Politiques

Contrôle du Trafic

Les Network Policies permettent de restreindre les accès entre les pods selon des règles spécifiques.

Règles de Communication

Créez un fichier YAML pour définir les règles de communication entre les pods.

Exemple de Policy

Autorise uniquement les pods "frontend" à communiquer avec les pods "backend".

Bonnes Pratiques pour le MCO

Quotas de Ressources

Limitez les ressources pour éviter que certains pods ne consomment toutes les ressources.

Alertes

Configurez des alertes pour être notifié en cas de défaillance.

Automatisation

Utilisez des solutions de CI/CD pour automatiser les déploiements et les mises à jour continues.