

REPUBLIQUE DEMONCRATIQUE DU CONGO

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET UNIVERSITAIRE

UNIVERSITE DE KINSHASA



FACULTE DES SCIENCES ET TECHNOLOGIES

DEUXIEME LICENCE EN MENTION INFORMATIQUE

**SUJET : INSTALLATION ET CONFIGURATION DE KUBERNETES
(K8s) SOUS UBUNTU**

1. PALANA ERANGER JOSUE
2. SEMBALA POLOTO KARIM
3. TENGE-TENGE MUZINGA PRÉCIEUX
4. MANITU LANDA LANDRY
5. NZUZI NTEDIKA ESPERAND
6. KABAMBA TSHIDIUPUILA JOSEPH
7. MUDIA KIMPANI WILLY
8. LIZINATA KONGO DIEU-MERCI
9. GBOZE KAZONGO ELDEN
10. MANUELLA MULUBA GABRIELA

ANNEE ACADEMIQUE 2025 - 2026

Installation et Configuration de Kubernetes sous Ubuntu

Kubernetes est aujourd’hui l’un des outils les plus utilisés pour l’orchestration des conteneurs dans les environnements informatiques modernes. Ce rapport présente de manière détaillée l’installation et la configuration de Kubernetes sur le système d’exploitation Ubuntu. L’objectif principal est de mettre en place un cluster Kubernetes fonctionnel, en expliquant les concepts fondamentaux, les prérequis techniques, les étapes d’installation ainsi que la configuration de base permettant le déploiement et la gestion d’applications conteneurisées. Ce travail s’inscrit dans un cadre académique et vise à fournir une compréhension claire et pratique de Kubernetes.

MOTS-CLÉS

Kubernetes, Docker, Orchestration, Cluster, Pods, Noeuds

INTRODUCTION

Avec l’évolution rapide des technologies informatiques, la conteneurisation est devenue une solution incontournable pour le déploiement et la gestion des applications. Docker a permis de simplifier l’exécution des applications dans des environnements isolés, mais la gestion de plusieurs conteneurs à grande échelle reste complexe. Kubernetes, développé initialement par Google, apporte une solution efficace à ce problème en assurant l’orchestration automatique des conteneurs.

L’objectif de ce rapport est de présenter les différentes étapes nécessaires à l’installation et à la configuration de Kubernetes sous Ubuntu, tout en expliquant les concepts essentiels liés à son fonctionnement.

CHAPITRE I : GÉNÉRALITÉS SUR KUBERNETES

1.1 Définition de Kubernetes

Kubernetes est une plateforme open source d'orchestration de conteneurs qui permet d'automatiser le déploiement, la mise à l'échelle et la gestion des applications conteneurisées. Il fonctionne sous forme de cluster composé de plusieurs machines appelées nœuds.

1.2 Rôle de Kubernetes

Kubernetes assure notamment :

- La gestion automatique des conteneurs
- La haute disponibilité des applications
- La répartition de charge
- La mise à l'échelle automatique
- La tolérance aux pannes

1.3 Architecture de Kubernetes

Un cluster Kubernetes est composé de deux types de nœuds :

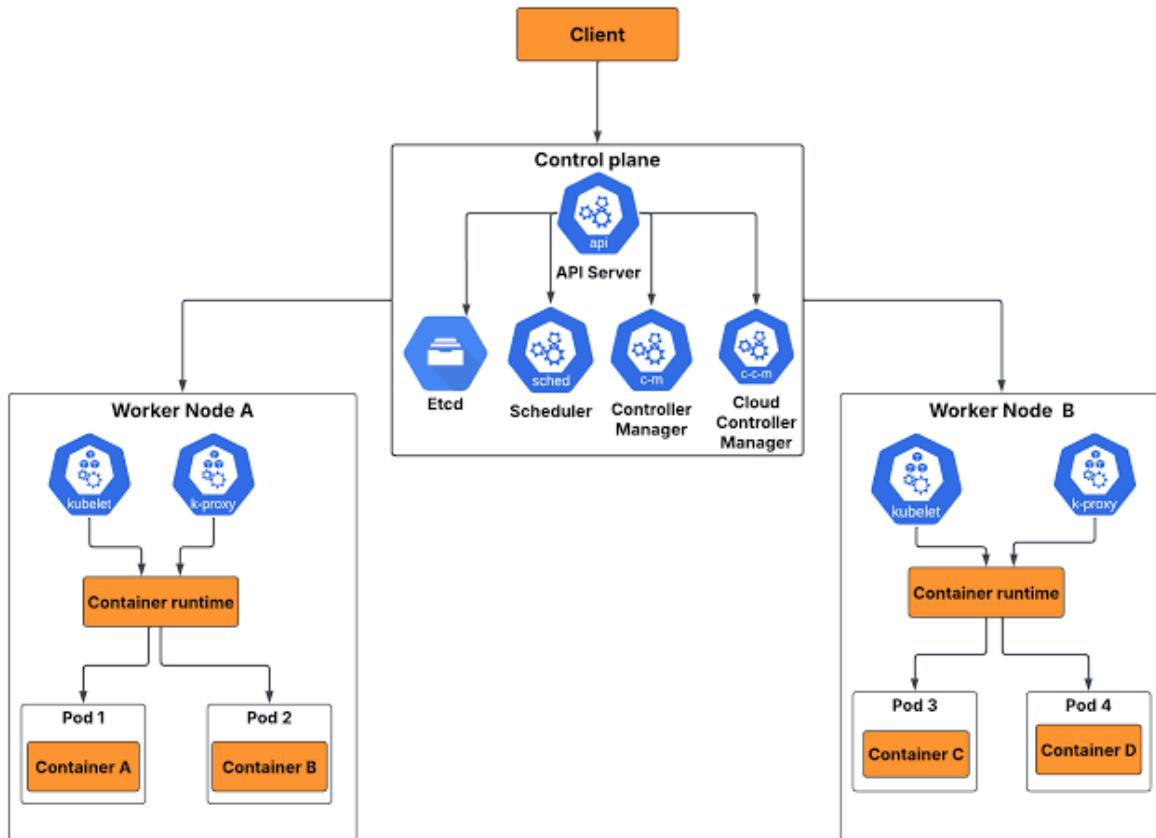
- Le Control Plane (Master) : responsable de la gestion globale du cluster
- Les Worker Nodes : responsables de l'exécution des applications



kubernetes

Logo de Kubernetes

ARCHITECTURE DE KUBERNETES



🌐 Vue d'ensemble du schéma

Ce schéma représente l'architecture d'un cluster Kubernetes.
Un cluster Kubernetes est composé de deux types de nœuds :

1. Control Plane (le cerveau du cluster)
2. Worker Nodes (les ouvriers qui exécutent les conteneurs)

Et tout part du Client (toi ou ton navigateur, ou kubectl).

1 Client

- C'est quoi ? : Celui qui interagit avec le cluster.
Par exemple : ton terminal avec kubectl, ou une application web.
- Rôle : envoyer des commandes pour créer, gérer ou supprimer des applications.

2 Control Plane (plan de contrôle)

C'est le cerveau du cluster Kubernetes.
Il décide quoi faire, où le faire, et quand le faire.

Il est composé de plusieurs composants :

a) API Server

- Cœur du control plane, point d'entrée du cluster.
- Rôle : reçoit toutes les requêtes du client et communique avec le cluster.

b) Etcd

- Base de données du cluster.
- Rôle : stocke l'état actuel du cluster (qui fait quoi, où, combien de pods...).
- Exemple : "Le Deployment hello-world doit avoir 1 Pod en cours d'exécution"

c) Scheduler

- Planificateur, décide sur quel worker node mettre les pods.

d) Controller Manager

- Superviseur automatique, s'assure que l'état réel correspond à l'état désiré.

Exemple : Si un Pod meurt, il le recrée automatiquement.

e) Cloud Controller Manager

- Gère l'intégration avec le cloud si tu es sur AWS, GCP, Azure.

- Sur Minikube, il n'est pas très utilisé, mais en production, il configure le réseau ou le stockage.

3 Worker Nodes (les ouvriers)

Ce sont les machines qui exécutent les applications (Pods / conteneurs). Chaque worker node contient :

a) kubelet

- Agent qui communique avec le control plane.
- Rôle : dit "je vais exécuter ce Pod ici" et informe le control plane.

b) kube-proxy

- Gère le réseau et l'accès au Pod.
- Expose les services et permet aux Pods de communiquer.

c) Container runtime

- Moteur de conteneurs, par ex. Docker ou containerd.
- C'est ce qui exécute réellement le conteneur.

4 Pods et Containers

- Pod = unité de déploiement. Contient 1 ou plusieurs conteneurs.
- Container = l'application réelle qui tourne.

5 Flux de communication (simplifié)

1. Client demande un déploiement
2. API Server reçoit la demande
3. Scheduler choisit le Worker Node approprié
4. Controller Manager crée le Pod
5. kubelet sur le Worker Node exécute le Pod via le container runtime
6. kube-proxy expose le Pod pour que le réseau puisse y accéder
7. Pod exécute le container → ton application tourne

CHAPITRE II : PRÉREQUIS POUR L'INSTALLATION

2.1 Prérequis matériels

- Processeur : 2 cœurs minimum
- Mémoire RAM : 2 Go minimum (4 Go recommandé)
- Espace disque : 20 Go minimum

2.2 Prérequis logiciels

- Système d'exploitation : Ubuntu 20.04 ou 22.04
- Accès administrateur (sudo)
- Connexion Internet stable
- Désactivation du swap (obligatoire pour Kubernetes)

CHAPITRE III : INSTALLATION DE KUBERNETES SOUS UBUNTU

3.1 Mise à jour du système

La première étape consiste à mettre à jour le système afin d'éviter tout conflit de dépendances.

```
sudo apt update  
sudo apt upgrade -y
```

3.2 Installation du moteur de conteneurs Docker

```
Sudo apt install Docker.io -y
```

```
Docker --version
```

```
Sudo usermod -Ag docker $USER (autoriser ton utilisateur à utiliser  
Docker)
```

3.3 Installation de Minikube

```
Curl -LO
```

```
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

```
Sudo install minikube-linux-amd64 /usr/local/bin/minikube  
Minikube version
```

3.4 Installer Kubectl (Outil Kubernetes)

```
Curl -LO "https://dl.k8s.io/release/$(curl -L -S https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
Sudo install kebectl /usr/local/bin/kubectl
```

```
Kubectl version --client
```

3.5 Démarrage et vérification du fonctionnement

```
Minikube start --driver=docker
```

À cette étape un cluster est créé et puis vérifier si toute la configuration fonctionne

- Vérifier le cluster

```
Kubectl get nodes
```

- Vérifier les pods système

```
Kubectl get pods -A
```

CONCLUSION

Ce rapport a présenté de manière détaillée l'installation et la configuration de Kubernetes sous Ubuntu. À travers les différentes étapes décrites, il a été démontré que Kubernetes constitue une solution puissante pour l'orchestration des conteneurs. Malgré sa complexité initiale, son adoption offre des avantages considérables en termes de fiabilité, de performance et de gestion des applications modernes.