

Voici le lien pour votre projet K8S

<https://cloud-hb.notion.site/3k8S-Project-1d57107ac6bc80279420e8b859baa905>

## **3k8S Project**

### **Projet Kubernetes: Déploiement d'une infrastructure web multi-environnements**

#### **Contexte**

L'entreprise TechSolutions souhaite moderniser son infrastructure et adopter Kubernetes pour déployer ses applications. En tant que nouvelle équipe DevOps, vous êtes chargés de créer une preuve de concept qui démontre les capacités de Kubernetes pour gérer différents environnements de déploiement.

Le CTO vous a fourni des applications conteneurisées simples (un frontend, une API backend et une base de données) et vous demande de concevoir une infrastructure Kubernetes qui permettra de les déployer de manière isolée dans des environnements de développement, de test et de production, tout en assurant leur surveillance et leur sécurité.

#### **Objectifs du projet**

1. Mettre en place un cluster Kubernetes on-premise
2. Créer une architecture multi-environnements avec isolation
3. Déployer les applications dans chaque environnement
4. Configurer l'accès sécurisé et les contrôles de ressources
5. Installer un système de monitoring basique
6. Mettre en œuvre différentes stratégies de déploiement
7. Préparer une démonstration complète du système

#### **Travail à réaliser**

##### **Partie 1 : Installation et configuration du cluster (1 semaine)**

1. Installer Minikube sur votre machine de développement avec au moins 4 GO de RAM et 2 CPU
2. Activer les addons nécessaires : metrics-server, dashboard
3. Configurer kubectl pour interagir avec votre cluster
4. Installer le dashboard Kubernetes pour visualiser les ressources
5. Documenter précisément votre configuration avec des captures d'écran et les commandes utilisées

**Livrable attendu** : Document décrivant l'installation avec les commandes utilisées et les captures d'écran de vérification du bon fonctionnement.

##### **Partie 2 : Création d'une architecture multi-environnements**

1. Créer trois namespaces distincts avec des configurations spécifiques :

- dev : pour le développement (ressources limitées)
  - staging : pour les tests (ressources moyennes)
  - prod : pour la production simulée (ressources prioritaires)
2. Pour chaque namespace, déployer :
- Un déploiement MySQL avec :
    - Un volume persistant pour les données
    - Un secret pour le mot de passe root
    - Un ConfigMap pour les paramètres de configuration
    - Un service pour l'exposer uniquement en interne
  - Un déploiement pour l'API backend avec :
    - Une configuration pour se connecter à la base de données correspondante
    - Des readiness et liveness probes pour vérifier son bon fonctionnement
    - Un service pour l'exposer
  - Un déploiement pour le frontend avec :
    - Une configuration pour se connecter à l'API backend
    - Un service pour l'exposer à l'extérieur
3. S'assurer que les applications peuvent communiquer entre elles uniquement dans leur namespace respectif
4. Configurer des variables d'environnement différentes pour chaque environnement via des ConfigMaps
5. Mettre en place des limites de ressources différentes selon l'environnement à l'aide de LimitRange

**Livrable attendu** : Fichiers YAML de configuration pour tous les déploiements, services, ConfigMaps, secrets et PersistentVolumes, ainsi qu'un schéma d'architecture montrant comment les composants interagissent.

### Partie 3 : Sécurité et contrôle d'accès

1. Créer trois comptes utilisateurs avec différents niveaux d'accès :
  - admin-user : accès complet à tous les namespaces
  - dev-user : accès en lecture/écriture uniquement au namespace dev
  - viewer-user : accès en lecture seule à tous les namespaces
2. Configurer les Roles, ClusterRoles, RoleBindings et ClusterRoleBindings nécessaires
3. Mettre en place des NetworkPolicies (activer le CNI approprié dans Minikube) pour :

- Isoler les bases de données (accessible uniquement par les backends du même namespace)
  - Limiter l'accès aux backends (accessibles uniquement par les frontends du même namespace)
  - Permettre l'accès externe uniquement aux frontends
4. Configurer des ResourceQuotas pour chaque namespace afin de limiter :
- Le nombre total de pods
  - La consommation de CPU et mémoire
  - Le nombre de services, secrets et configmaps

**Livrable attendu :** Fichiers YAML pour la configuration RBAC et les NetworkPolicies, documentation des tests effectués pour vérifier que les restrictions fonctionnent correctement.

#### **Partie 4 : Déploiements avancés et monitoring**

1. Configurer deux stratégies de déploiement différentes :
  - Rolling update pour l'environnement de production (max 25% indisponible à la fois)
  - Recreate pour l'environnement de développement
2. Mettre en place un HorizontalPodAutoscaler pour le backend en production :
  - Scaling basé sur l'utilisation CPU (> 70%)
  - Minimum 2 pods, maximum 5 pods
  - Créer un script simple pour générer de la charge sur l'API
3. Installer Prometheus et Grafana dans un namespace monitoring :
  - Configurer Prometheus pour collecter les métriques des pods et des nodes
  - Créer un tableau de bord Grafana pour visualiser :
    - L'utilisation CPU et mémoire par namespace
    - Le nombre de pods par déploiement
    - Le statut des liveness/readiness probes
    - Les temps de réponse des services
4. Configurer des services NodePort pour exposer les applications frontend :
  - Exposer chaque frontend sur un port différent
  - Documenter comment accéder aux différentes applications
  - Expliquer les différences entre les types de services (ClusterIP, NodePort, LoadBalancer)

**Livrable attendu :** Configuration complète du monitoring, captures d'écran des tableaux de bord, fichiers YAML pour les services NodePort et les HPA, documentation sur les stratégies de déploiement.

### **Livrables finaux pour la soutenance**

1. Ensemble des fichiers YAML organisés dans une structure de répertoires logique
2. Document d'architecture technique expliquant :
  - L'architecture globale (avec schémas)
  - Les choix techniques et leurs justifications
  - Les difficultés rencontrées et les solutions apportées
  - Les perspectives d'amélioration
3. Présentation pour la soutenance
4. Démonstration en direct incluant :
  - Vue d'ensemble du cluster avec le dashboard
  - Déploiement d'une nouvelle version de l'application dans les différents environnements
  - Simulation d'une panne et observation de l'auto-réparation
  - Vérification des restrictions d'accès avec différents utilisateurs
  - Visualisation des métriques dans Grafana lors d'un test de charge
  - Démonstration du scaling automatique

### **Remarques importantes**

- **Vous n'avez pas besoin de développer les applications** - utilisez des images Docker existantes (nginx, mysql, et une image simple pour l'API comme httpd avec un message personnalisé)
- Concentrez-vous sur la configuration Kubernetes et non sur le code des applications
- Tous les éléments doivent pouvoir communiquer correctement dans leur namespace respectif
- La base de données doit être accessible uniquement par le backend, qui lui-même est accessible par le frontend
- Documentez clairement les ports utilisés et les variables d'environnement nécessaires
- Préparez votre démonstration à l'avance et testez-la pour éviter les surprises
- Créez des scripts ou des commandes préparées pour faciliter la démonstration en direct

### **Critères d'évaluation**

- Fonctionnalité de l'infrastructure (40%)
- Compréhension et application des concepts Kubernetes (30%)

- Qualité de la documentation et des schémas (15%)
- Présentation et démonstration (15%)

#### **Ressources recommandées**

- Documentation officielle de Kubernetes
- Documentation de Minikube
- Tutoriels sur Prometheus et Grafana
- Guide sur les NetworkPolicies et RBAC
- Exemples de configurations pour les services utilisés

Bonne chance pour ce projet qui vous permettra d'acquérir des compétences concrètes et très demandées sur le marché du travail !

[Binômes de Soutenance – Projet Kubernetes](#)



# Binômes de Soutenance – Projet Kubernetes

1. TCHAKOUTE FAGNA Francki Leonel + MENGUE MENDOUGA José Astrid
2. NJOUNGANG KINKEU Igor Landry + SEUTIO FOKOU Gélande
3. MONTHE Modeste Landry + NDJONGUE TIADEU Malika Lucretse
4. ZANG Laurent + KAMGA FOTSO Franck Ulrich
5. FANKEM Marx Marvel + TADEFO Ghislain Arnaud
6. WAFO NDE Michaeli Ezechiel + KWOKAM TATOU Ronald Steve
7. MANDENG Danielle Audrey + BOUGNA FOTIO Sinclair
8. AOULAD TOUHAMI Ismail + BELAIDI Maria
9. WAMBO NONO Bryan + NGANGOYA NZOKOU Wilfried Stive
10. NZOMBOU De Chardin + NJEUHOUE TCHAMDJOU Orline Vanelle
11. NGNIPIBEO Jules Loïc + OUEDRAOGO Ramata Dite Martine
12. KENFACK KAMLEU Emmerencia Patricia + KAMDEM FOAKIM Lionel Stephane
13. FANKEP POUEME Saint-Ines + LONTCHI KENFACK Merveille
14. BALDE Fatoumata Binta + TCHAPDA MBE Nelson
15. GANGUE MBOUOM Maxime Harold + Rabat + Rollin Pavel
16. KEMTA SA'A Luc + NJOH BOYOMO Thérèse Carla Emmanuella

Bon courage,