

# RÉPARTITION DES TÂCHES PROJET OS

## Titre: Cluster Kubernetes

Description: Déploiement d'un chatbot complet avec une architecture frontend-backend-base de données sur un cluster Kubernetes. L'application comprendra un frontend, un backend API, et une base de données. Utilisation de Kubernetes pour orchestrer et gérer les conteneurs.

### **Étudiant A** — Backend + API OpenAI + Kubernetes backend

Rôle général : Construction du cerveau applicatif.

#### 1. Développement backend

- développe l'API (Python/Node)

- endpoints :

  - \* POST /chat

  - \* GET /history/:user

  - \* POST /new-conversation

- gestion du prompt système

- gestion du format streaming

#### 2. Intégration OpenAI

- appel à l'API OpenAI

- gestion des erreurs

- sécurisation de la clé via Secret Kubernetes

#### 3. Persistance

- connexion à la DB fournie par l'étudiant C

- stockage historique

#### 4. Dockerisation

- Dockerfile backend
- tests locaux

#### 5. Kubernetes (Kind local)

- backend-deployment.yaml
- backend-service.yaml
- configMap
- secret mocké

#### 6. Documentation

- README backend
- schéma API
- notes d'installation

#### **Livrables**

backend fonctionnel + image Docker backend + manifests Kubernetes backend testés sous Kind + documentation backend

### **Étudiant B** — Frontend Web + UI de chat + Kubernetes frontend

Rôle général : Construire l'interface utilisateur.

#### 1. Développement Frontend

- UI moderne style ChatGPT
- historique

- appels API backend
- streaming si activé
- responsive et dark mode

## 2. Intégration API backend

- POST /chat
- récupération de l'historique

## 3. Dockerisation

- Dockerfile frontend
- tests locaux

## 4. Kubernetes (Kind local)

- frontend-deployment.yaml
- frontend-service.yaml
- configMap
- ingress local optionnel

## 5. Documentation

- installation
- schéma UI
- endpoints utilisés

### **Livrables**

UI complète + image Docker frontend + manifests Kubernetes frontend testés sous Kind  
+documentation frontend

## Étudiant C — Infrastructure Kubernetes + Base de données + Ingress

Rôle général : Le socle technique.

### 1. Cluster

- installation cluster (Kind ou VM)
- namespace chatbot-app

### 2. Base de données

- choix DB (Postgres/Mongo)
- Deployment/StatefulSet
- Service interne
- PVC
- Secret identifiants

### 3. Ingress Controller

- Installation NGINX Ingress Controller
- règles ingress backend+frontend
- HTTPS optionnel

### 4. Gestion des secrets

- clé OpenAI
- DB credentials

### 5. Intégration finale

- récupère les manifests A et B
- déploie

- tests communication

## 6. Monitoring (optionnel)

- Prometheus
- Grafana

## **Livrables**

cluster complètement fonctionnel + DB opérationnelle + ingress configuré + documentation complète du cluster

## **Tableau récapitulatif**

Étudiant A : Backend + API + Kubernetes applicatif

Étudiant B : Frontend + UI + Kubernetes applicatif

Étudiant C : Infra + DB + Ingress