

## Cahier des charges pour le miniprojet CaaS

Objectif du projet : Le but de ce projet est de mettre en place une chaîne d'intégration et de déploiement continu pour une application, en utilisant des outils couramment utilisés dans le monde DevOps, notamment GitHub/GitLab, Docker, Jenkins et Kubernetes.

### 1. Référentiel de code source:

- Créer un référentiel pour l'application soit sur GitHub soit sur GitLab.
- Ce référentiel doit être rendu public à la fin du projet pour faciliter l'évaluation.

### 2. Dockerisation de l'application:

- Réaliser un Dockerfile permettant de construire une image Docker de l'application.
- L'application contenue dans cette image doit être pleinement fonctionnelle une fois le conteneur lancé.

### 3. Mise en place de la pipeline Jenkins:

- Installer et configurer Jenkins.
- Créer une pipeline Jenkins avec les étapes suivantes :
  - Récupération du code source à partir du référentiel GitHub/GitLab.
  - Construction de l'image Docker à partir du Dockerfile.
  - Pousser cette image sur Docker Hub (assurez-vous d'avoir un compte Docker Hub).
  - Mise à jour du déploiement via GitOps, si possible.

### 4. Kubernetes (k8s):

- Créer au moins un objet de déploiement pour l'application.
- Créer un objet de service pour exposer l'application.
- Les objets k8s doivent être déployés via l'un des outils suivants : kubectl, helm ou argocd.

### 5. Monotoring (Prometheus + Grafana):

- Mettre en place une solution de monitoring basée sur **Prometheus + Grafana** (ex : via Helm chart kube-prometheus-stack).
- Exigences minimales :
  - Accès à **Grafana** (port-forward ou Ingress).
  - Vérification que Prometheus collecte les métriques du cluster (nodes, pods, etc.).
  - Création ou import d'au moins **un dashboard Grafana** pertinent (cluster / workloads).
  - Ajout d'au moins **un indicateur applicatif** ou une métrique liée au service (ex : métriques HTTP, latence, erreurs), si possible.
- Bonus (facultatif) :
  - Mise en place d'une **alerte simple** (Alertmanager) ou d'une règle Prometheus.

## 6. Livrables:

- Lien vers le référentiel GitHub/GitLab public.
- Dockerfile et autres scripts ou fichiers de configuration utilisés.
- Instructions détaillées pour reproduire l'environnement et la chaîne CI/CD.
- Document explicatif (Readme).

**Évaluation:** Le projet sera évalué sur les critères suivants :

- Respect des exigences du cahier des charges.
- Utilisation des connaissances vues dans les TPs.
- Qualité du code et des configurations.
- Fonctionnalité de la chaîne CI/CD.
- Choix de l'architecture de l'application (**monolithique** ou **microservices**)

**Notification importante:** Vos projets doivent impérativement être remis sur **Moodle** !  
Toute autre remise, comme par *email*, ne sera pas comptabilisé!