

Cahier des Charges – Projet M1

Mastère DevOps

Titre : Mise en place d’une plateforme CI/CD et d’un environnement DevOps industrialisé

I. Contexte et problématique

Le client est une entreprise de services numériques (ESN) spécialisée dans le développement d’applications web pour des clients PME et collectivités locales.

Son modèle économique repose sur la capacité à livrer rapidement de nouvelles versions d’applications, avec un niveau de qualité constant.

Problème actuel :

- Les builds et déploiements sont manuels et varient selon les équipes.
- Les tests ne sont pas systématisés, entraînant des bugs en production.
- Aucun monitoring centralisé ne permet de savoir si un service est surchargé, instable ou énergivore.
- Les environnements sont hétérogènes (VM, containers isolés, machines physiques), rendant difficile la scalabilité.
- La sécurité des livraisons est insuffisante (pas de scans automatisés, gestion des secrets artisanale).

La direction souhaite professionnaliser ses pratiques grâce à une plateforme DevOps complète, centralisant le cycle de vie applicatif.

II. Objectifs du projet

- Réduire le temps de mise en production de 30 %.
- Fiabiliser les livraisons (moins d’incidents liés aux déploiements).
- Améliorer la supervision (indicateurs métiers + techniques en temps réel).
- Sécuriser la chaîne logicielle (scans, secrets, conformité).
- Réduire l’empreinte carbone de l’infrastructure (Green IT).

III. Organisation du client et besoins par pôle

Pôle	Besoins spécifiques	Exemples de solutions
Développement	Automatiser les builds, tests et livraisons	GitLab CI/CD, Jenkins, SonarQube
Exploitation (Ops)	Déploiements rapides, rollback simple, scalabilité	Docker, Kubernetes, Helm
Sécurité	Analyse de vulnérabilités, gestion des secrets, conformité RGPD	Trivy, Vault, Dependency-Check
Direction	Visibilité sur les performances et coûts	Grafana (dashboards métiers + Green IT)

IV. Fonctionnalités attendues

1. Chaîne CI/CD complète :

- Build automatisé, exécution des tests, scan de sécurité, déploiement.
- Gestion des versions (ex: branches, releases, rollback).

2. Conteneurisation et orchestration :

- Conteneurisation des applications avec Docker.
- Déploiement orchestré (ex: Kubernetes, K3s ou Minikube).

3. Supervision & monitoring :

- Collecte des logs et métriques.
- Tableaux de bord temps réel.
- Alerting en cas de problème.

4. Sécurité intégrée :

- Scans automatiques de vulnérabilités (images, dépendances).
- RBAC (contrôle d'accès par rôle).
- Gestion sécurisée des secrets (Vault).

5. Optimisation énergétique (Green IT) :

- Auto-scaling en fonction de la charge.
- Analyse de la consommation CPU/RAM.
- Optimisation des pipelines pour limiter la surconsommation.

6. Documentation complète :

- Architecture technique.
- Guide utilisateur et guide d'administration.

7. (Bonus) : introduction au chaos engineering ou démonstration de résilience (panne simulée).

V. Solution technique conseillée

Stack recommandée :

Domaine	Outils conseillés
CI/CD	GitLab CI/CD ou Jenkins
Conteneurisation	Docker
Orchestration	Kubernetes / K3s / Minikube
Monitoring	Prometheus + Grafana
Logs	ELK (Elasticsearch, Logstash, Kibana)
Sécurité	Trivy, Clair, HashiCorp Vault
Infrastructure as Code	Terraform, Ansible
Stockage artefacts	MinIO (compatible S3)

Workflows type :

1. Développeur pousse du code sur Git.
2. CI/CD pipeline → build + tests + scan sécurité.
3. Image Docker générée → stockée dans un registry.
4. Déploiement automatique dans Kubernetes.
5. Monitoring → alertes envoyées en cas d'échec.

VI. Innovations attendues

- Industrialisation : pipelines reproductibles et modulaires.
- Sécurité by design : intégration de scans dès la CI/CD.

- Green IT : suivi énergétique + auto-scaling.
- Observabilité complète : logs + métriques + alertes.
- Interopérabilité : intégration possible avec d'autres SI (API, Slack/Teams).

VII. Répartition conseillée des rôles

Exemple de répartition des rôles

Étudiant	Rôle principal	Missions
Étudiant 1	Responsable CI/CD	Configuration pipelines, automatisation des builds, tests, scans
Étudiant 2	Responsable Conteneurs & Orchestration	Création des images Docker, orchestration avec Kubernetes
Étudiant 3	Responsable Monitoring & Sécurité	Mise en place Prometheus/Grafana, gestion logs, intégration scans vulnérabilités
Étudiant 4 (optionnel)	Coordinateur & Documentation	Supervision projet, rapport technique, guide utilisateur, soutenance

VIII. Livrables attendus

- Analyse du besoin.
- Architecture technique : schéma, justification des choix.
- MVP fonctionnel : pipeline complet + déploiement conteneurisé.
- Tableaux de bord : monitoring et sécurité.
- Rapport technique : architecture, pipeline, tests, sécurité, Green IT.
- Guide utilisateur et d'administration.
- Vidéo de démonstration claire et professionnelle

IX. Aide complémentaire et bonnes pratiques

Ressources disponibles :

- GitHub Student Pack : GitHub Pro, JetBrains, MongoDB, hébergement cloud.

Bonnes pratiques :

- Respect du RGPD et du cadre légal.
- Documentation continue.
- Pipelines reproductibles et sécurisés.
- Tests automatisés intégrés.
- Suivi Green IT (consommation et optimisation).
- KPIs minimums à suivre :
 - Temps moyen de déploiement ;
 - Taux de réussite des builds ;
 - Nombre de vulnérabilités détectées ;
 - Consommation CPU/RAM par environnement.