



PROPOSITION DE SUJET

## Projet de Fin d'Études

Master 2 DevOps

# FieldOps360

*Plateforme de Suivi des Équipes d'Intervention Terrain*

**Secteurs :** BTP, Électricité, Travaux Publics

**Durée :** 6 mois

**Architecture :** Multi-tenant, Microservices, Cloud-Native

# Table des Matières

Table des Matières .....	2
1. Contexte du Projet.....	3
2. Objectifs Pédagogiques .....	3
3. Architecture Fonctionnelle .....	3
3.1 Profils Utilisateurs.....	3
3.2 Modules Fonctionnels .....	3
Module 1 : Gestion des Projets/Chantiers .....	3
Module 2 : Gestion des Ressources.....	4
Module 3 : Planification et Exécution.....	4
Module 4 : Traçabilité et Preuves.....	4
Module 5 : Reporting et Analytics .....	4
4. Architecture Technique (Focus DevOps) .....	5
4.1 Stack Technologique Proposée.....	5
4.2 Architecture Multi-Tenant.....	5
4.3 Observabilité.....	5
5. Périmètre DevOps (Travail Principal) .....	6
5.1 Infrastructure as Code .....	6
5.2 Pipeline CI/CD .....	6
5.3 Automatisation Multi-Tenant.....	6
6. Planning Prévisionnel (6 mois) .....	6
7. Livrables Attendus .....	7
7.1 Livrables Techniques .....	7
7.2 Documentation.....	7
7.3 Démonstration.....	7
8. Axes d'Innovation (Bonus).....	7
9. Compétences Développées .....	7
10. Encadrement et Modalités .....	8
10.1 Encadrement Entreprise.....	8
10.2 Attendus de l'Étudiant.....	8

# 1. Contexte du Projet

Dans le secteur du BTP, des travaux publics et des interventions techniques, la gestion des équipes terrain reste un défi majeur. Les entreprises manquent souvent d'outils adaptés pour suivre en temps réel l'avancement des chantiers, optimiser l'allocation des ressources et garantir la traçabilité des interventions.

Ce projet vise à développer une plateforme complète permettant de digitaliser et d'optimiser la gestion des interventions sur terrain, tout en offrant une solution commercialisable en mode SaaS multi-tenant.

# 2. Objectifs Pédagogiques

Ce projet permet à l'étudiant de mettre en pratique l'ensemble des compétences DevOps acquises durant sa formation :

Compétence DevOps	Application dans le projet
CI/CD	Pipeline automatisé multi-environnement
Containerisation	Architecture microservices Docker/Kubernetes
IaC	Terraform/Ansible pour provisioning
Monitoring	Stack observabilité complète
Multi-tenancy	Isolation et déploiement automatisé par client

# 3. Architecture Fonctionnelle

## 3.1 Profils Utilisateurs

La plateforme doit supporter différents profils avec des droits et des vues adaptés à leurs responsabilités :

Profil	Fonctionnalités principales
Manager	Dashboard KPI global, vue consolidée multi-sites, rentabilité par projet, alertes et notifications stratégiques
Chef de Projet	Planning Gantt interactif, affectation des ressources, suivi budgétaire, rapports d'avancement
Chef de Chantier	Plan de travail quotidien, validation des tâches, gestion des aléas, pointage équipe
Membre d'Équipe	Liste des tâches (mobile), upload photos/preuves, pointage géolocalisé, signature électronique

## 3.2 Modules Fonctionnels

### Module 1 : Gestion des Projets/Chantiers

- Création et paramétrage des chantiers (BTP, électrique, TP)

- Définition des phases et jalons
- Documents associés (plans, permis, etc.)

## Module 2 : Gestion des Ressources

- Parc véhicules (disponibilité, affectation, suivi GPS)
- Matériel et outillage (inventaire, réservation, maintenance)
- Matières premières (stock, commandes, consommation)

## Module 3 : Planification et Exécution

- Planning des interventions avec drag & drop
- Affectation automatique selon compétences/disponibilité
- Check-lists personnalisables par type de travaux

## Module 4 : Traçabilité et Preuves

- Horodatage et géolocalisation des actions
- Capture photo avec filigrane (date, GPS, projet)
- Signature électronique client/intervenant
- Historique complet auditabile

## Module 5 : Reporting et Analytics

- KPI temps réel (productivité, retards, coûts)
- Rapports automatiques (journalier, hebdomadaire, mensuel)
- Export PDF/Excel pour facturation

## 4. Architecture Technique (Focus DevOps)

### 4.1 Stack Technologique Proposée

Couche	Technologies
Frontend Web	React ou Vue.js avec TypeScript
Frontend Mobile	React Native / Flutter (PWA en fallback)
API Gateway	Kong ou Traefik (routing, rate limiting, auth)
Backend	Node.js/NestJS ou Python/FastAPI (microservices)
Message Broker	RabbitMQ ou Apache Kafka
Base de données	PostgreSQL (données relationnelles)
Cache	Redis
Stockage fichiers	MinIO ou S3
Orchestration	Kubernetes
CI/CD	GitLab CI ou GitHub Actions

### 4.2 Architecture Multi-Tenant

**Stratégie d'isolation :** Database-per-tenant pour une isolation complète des données (RGPD compliant).

Le portail Super Admin permettra le provisioning de nouveaux tenants, la configuration par secteur d'activité (BTP, Électrique, TP), la gestion des licences et facturation, ainsi que le monitoring global de la plateforme.

### 4.3 Observabilité

Composant	Outil	Usage
Métriques	Prometheus + Grafana	Performance, SLA
Logs	ELK Stack / Loki	Centralisation, recherche
Traces	Jaeger / Zipkin	Debug distribué
Alerting	AlertManager + PagerDuty	Gestion des incidents

## 5. Périmètre DevOps (Travail Principal)

### 5.1 Infrastructure as Code

L'étudiant devra mettre en place une infrastructure complète définie en code, incluant des modules Terraform pour le cluster Kubernetes, les bases de données, le stockage et le réseau. L'organisation devra prévoir des environnements distincts (dev, staging, production) ainsi qu'un système de provisioning automatisé pour les nouveaux tenants.

### 5.2 Pipeline CI/CD

Le pipeline devra couvrir l'ensemble du cycle de vie du code, depuis le push jusqu'au déploiement en production. Les étapes principales incluent le build et linting, les tests (unitaires et intégration), le scan de sécurité (SAST), la construction des images Docker, le déploiement en environnements successifs, et les tests end-to-end avec stratégie Blue/Green pour la production.

### 5.3 Automatisation Multi-Tenant

Un script de provisioning automatisé devra permettre de créer un nouveau client en quelques minutes. Les actions automatisées incluent la création du namespace Kubernetes, le déploiement d'une base de données dédiée, la configuration DNS et certificats SSL, l'initialisation des données de référence, et la création du compte administrateur client.

## 6. Planning Prévisionnel (6 mois)

Mois	Phase	Livrables
M1	Cadrage et Architecture	Spécifications, choix technologiques, maquettes
M2	Infrastructure	IaC, cluster Kubernetes, CI/CD de base
M3	Backend Core	Services principaux, API, authentification multi-tenant
M4	Frontend et Mobile	Interfaces par profil, mode offline
M5	Intégration et Tests	Pipeline complet, tests automatisés, sécurité
M6	Production	Déploiement, monitoring, documentation technique

## 7. Livrables Attendus

### 7.1 Livrables Techniques

- Repository Git avec branching strategy (GitFlow)
- Infrastructure as Code (Terraform + Ansible)
- Helm Charts pour déploiement Kubernetes
- Pipeline CI/CD complet (GitLab CI / GitHub Actions)
- Stack de monitoring opérationnelle
- Scripts d'automatisation tenant provisioning
- Tests automatisés (unitaires, intégration, E2E)

### 7.2 Documentation

- Dossier d'architecture technique
- Guide de déploiement et d'exploitation
- Runbooks pour incidents courants
- Documentation API (OpenAPI/Swagger)

### 7.3 Démonstration

- Environnement de démo fonctionnel
- Scénario de provisioning nouveau client
- Dashboard de monitoring

## 8. Axes d'Innovation (Bonus)

L'étudiant pourra explorer selon son avancement les axes d'innovation suivants :

1. **GitOps** avec ArgoCD/Flux pour le déploiement déclaratif
2. **Service Mesh** (Istio/Linkerd) pour la communication inter-services
3. **Chaos Engineering** pour tester la résilience
4. **FinOps** - Optimisation des coûts cloud par tenant
5. **MLOps** - Prédiction de durée des tâches basée sur l'historique

## 9. Compétences Développées

Domaine	Compétences
<b>Infrastructure</b>	Kubernetes (CKA level), Terraform, Cloud Provider (AWS/GCP/Azure)
<b>CI/CD</b>	Pipeline design, GitOps, Release management
<b>Observabilité</b>	Prometheus/Grafana, Log management, Distributed tracing
<b>Sécurité</b>	RBAC & IAM, Secrets management, SAST/DAST
<b>SRE Practices</b>	SLI/SLO/SLA, Incident management, Capacity planning

## 10. Encadrement et Modalités

### 10.1 Encadrement Entreprise

**Encadrant :** CTO - SSII

- Points hebdomadaires de suivi
- Accès aux ressources cloud de l'entreprise
- Mentorat sur les pratiques DevOps industrielles

### 10.2 Attendus de l'Étudiant

- Autonomie dans la recherche de solutions
- Documentation continue du travail
- Présentation régulière des avancements
- Respect des bonnes pratiques (code review, tests)

#### Conclusion

Ce sujet offre un périmètre complet permettant à l'étudiant de démontrer sa maîtrise des pratiques DevOps tout en livrant un produit à valeur business réelle, potentiellement industrialisable par la SSII. Le projet couvre l'ensemble du spectre des compétences attendues d'un ingénieur DevOps : infrastructure, automatisation, monitoring, et pratiques SRE.