



DataPress – POC Conteneurisation & Kubernetes

Présentation pour
le DSI – Projet
Étudiant



Contexte DataPress

- Plateforme actuelle monolithique sur un seul serveur
- Risques : indisponibilité totale, MAJ risquées, tests impossibles
- Documentation faible et infrastructures vieillissantes
- Besoin de moderniser : Docker, Kubernetes, CI/CD

Objectifs du POC

- Séparer Front et API
- Mettre en place un environnement de recette fiable
- Utiliser Docker + Docker Compose en développement
- Déployer sur Kubernetes (namespace dédié)
- Introduire un premier workflow CI/CD



Architecture globale

- Front (Flask + Gunicorn)
↓ HTTP
- API (Flask + Gunicorn)
- Mode Dev : Docker Compose
- Mode Recette : Kubernetes

Architecture – Mode Développement

- Docker Compose :
- Service API → <http://localhost:8000>
- Service Front → <http://localhost:8080>
- Réseau interne Docker ('api' ↔ 'front')
- Dockerfiles multi-stage & utilisateurs non-root

Architecture — Kubernetes

- Namespace : datapress-recette
- API Deployment : 2 replicas
- Front Deployment : 1 replica
- Services :
 - API : ClusterIP
 - Front : NodePort
- ConfigMap + Secret
- Readiness & Liveness probes (/health)

Probes & Gestion des ressources

- Liveness probe : /health toutes les 20s
- Readiness probe : /health toutes les 10s
- Garantit redémarrage automatique et disponibilité
- Requests & Limits mémoire définis



CI/CD Minimal

- Workflow GitHub Actions :
 - Déclenché sur push
 - Checkout du code
 - Build de l'image Docker de l'API
 - Permet de valider le Dockerfile automatiquement
- (Difficultés rencontrées : GitHub Actions ne détectait pas le workflow)



8 Guide d'exploitation

- Mode Dev :
docker compose up --build
- Mode Recette :
kubectl apply -f k8s/
- Vérification :
*kubectl get all -n datapress-
recette*
kubectl logs <pod>

Limites & Améliorations

- Pas d'Ingress / TLS
- Pas de monitoring (Prometheus, Grafana...)
- Pas de HPA (autoscaling)
- Pas encore de registry + déploiement automatique
- Améliorations possibles pour la V2

Conclusion

- Ce POC démontre :
 - Séparation propre Front / API
 - Conteneurisation maîtrisée
 - Déploiement Kubernetes structuré
 - Exploitabilité simple
-
- Base solide pour moderniser la plateforme DataPress.