

UE Langages d'infrastructures LOGIN FISE A3

Introduction

2026-2027

RUE - Hélène Coullon <helene.coullon@imt-atlantique.fr>

et Lucas Gazeau (doctorant)



IMT Atlantique

Bretagne-Pays de la Loire

École Mines-Télécom

Table of Contents

1. “DevOps” - un concept
2. Cette UE DevOps
3. Présentation du projet

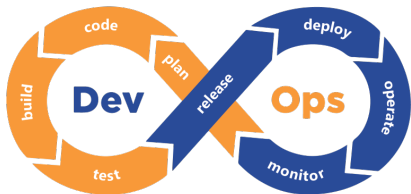
“DevOps” - un concept

Pourquoi le “DevOps” ?

- **Conflits d'objectifs** entre les développeurs et les opérateurs
 - Développer rapidement de nouvelles fonctionnalités
 - Avoir une infrastructure fiable et stable pour les clients
- **Métiers trop cloisonnés**
 - Le développeur ne sait pas déployer ses applications et administrer une infrastructure
 - L'opérateur ne connaît pas le fonctionnement et les besoins des applications qu'on lui donne à déployer
- Ces conflits ajoutent des **délais** conséquents voir des **blocages** pour fournir les nouvelles fonctionnalités aux utilisateurs !

L'ingénieur DevOps est là pour régler ces problèmes et faire le lien entre les développeurs et les opérateurs par le biais de **procédures automatisées**.

En quoi consiste ce concept ?



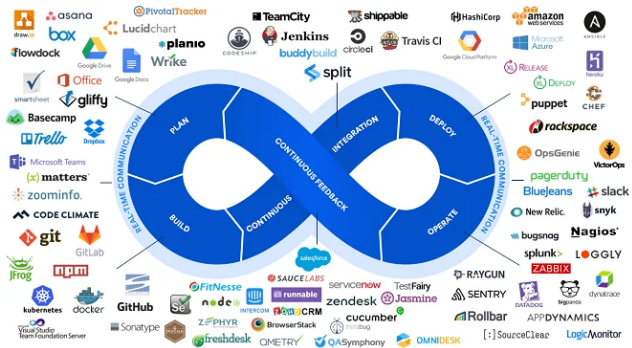
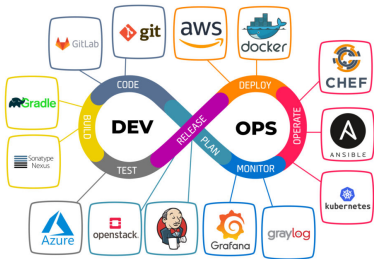
Le logo officiel du domaine schématise les objectifs d'un ingénieur DevOps.

Deux boucles y sont imbriquées :

- La boucle d'**intégration continue (CI)** : plan, code, build, test
- La boucle de **déploiement continu (CD)** : release, deploy, operate, monitor

L'ingénieur DevOps va devoir automatiser ces boucles pour accélérer le déploiement de nouvelles fonctionnalités à l'utilisateur

Mais qu'est-ce que c'est concrètement ?



En pratique, un ingénieur DevOps va utiliser tout un ensemble d'outils et de langages pour automatiser le processus et la fiabiliser !

Cette UE DevOps

Un métier complexe avec beaucoup de compétences



Code couleur

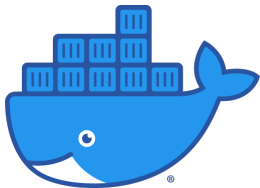
- déjà connu ?
- vu dans cette UE
- A3 / printemps
- ce qui sera laissé de côté

1. Concepts of development
2. OS & Linux
3. Networking and security
4. Containers
5. Automated CI/CD
6. Cloud providers
7. Containers orchestration
8. Monitoring
9. Infrastructure as Code
10. Scripting
11. Version control

Ce qui sera vu dans cette UE

3 modules :

1. Containers avec **Docker** et Docker-compose
2. Orchestration avec **Kubernetes**
3. Infrastructure as Code avec **Ansible**

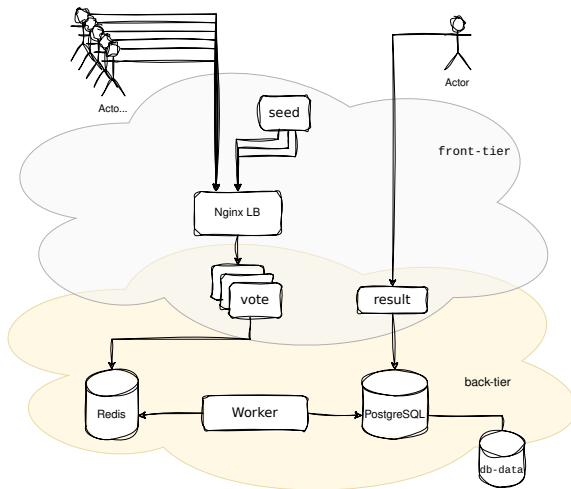


- BC3
 - Analyser une synthèse de plusieurs systèmes du cloud et du DevOps en répondant à des questions
 - Evaluations écrites
- BC5
 - Produire une solution de déploiement d'une application micro-services de bout en bout depuis l'approvisionnement de ressources jusqu'à la réparation et l'ajustement automatique
 - Oral et codes des projets

Présentation du projet

Projet de départ (Docker)

Conteneuriser et déployer localement cette application avec Docker



Objectifs

- Comprendre comment faire évoluer l'application et son déploiement si on utilise Kubernetes
- Ecrire les manifests Kubernetes nécessaires

Objectif

- Le contexte est de ne plus utiliser les stockages de Kubernetes
- Nous allons déployer une base de données distribuée sur une autre infrastructure (OpenStack) en utilisant Ansible

- Regarder la vidéo proposée
 - <https://www.youtube.com/watch?v=0yWAtQ6wYNM>
- Installations :
 - Linux :)
 - WSL2 (Windows)
 - Docker desktop
 - Kubectl et Minikube
 - Ansible