

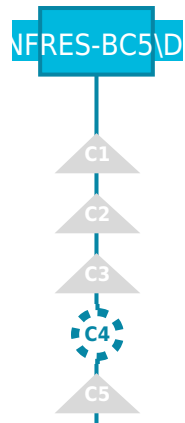
Pourquoi cette UE ?

Ce module est conçu pour répondre à plusieurs enjeux importants dans le développement logiciel moderne. Face à la complexité croissante des applications, les architectures traditionnelles monolithiques évoluent vers une application composée de petits services indépendants, plus faciles à gérer, à développer et à déployer (micro-service). L'intégration et le déploiement continus (CI/CD) permettent d'automatiser le processus de construction, de test et de déploiement de ces applications, ce qui accélère le cycle de développement et réduit les erreurs.

Éléments constitutifs de l'UE

	coefficient
INFRES_7_4_DL-1 Architecture logicielle: Microservices - 1	3
INFRES_7_4_DL-2 Intégration et déploiement continus	2
Volume d'heures d'enseignement encadré	Volume d'heures de travail personnel
50	14
Nombre d'ECTS	
4	

Alignement curriculaire

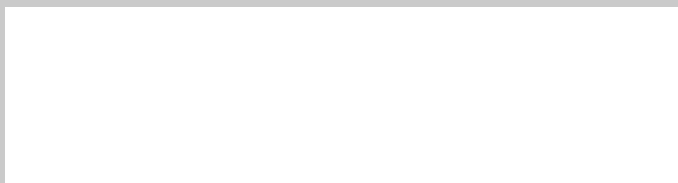
Parmi les compétences visées par la formation, lesquelles sont développées dans cette UE ?

BC1	L'UE ne contribue pas à ce bloc de compétences
BC1	L'UE contribue à ce bloc de compétences
C1	Compétence non adressée dans cette UE
C1	Compétence mise en œuvre dans cette UE
C1	Compétence enseignée dans cette UE
C1	Compétence évaluée dans cette UE
C1	Compétence enseignée et évaluée dans cette UE

Contexte et enjeux de l'enseignement

Ce cours vise à introduire un modèle d'architecture devenu incontournable avec le cloud et les applications distribuées. Il permet aux élèves de comprendre pourquoi on remplace progressivement les architectures monolithiques par des ensembles de services indépendants, interconnectés et facilement déployables. Les enjeux sont d'appréhender les bénéfices en termes de modularité, scalabilité, résilience et rapidité de mise à jour, mais aussi de saisir les défis associés : gestion des communications, orchestration, sécurité et supervision. Ce cours pose les bases conceptuelles et pratiques pour concevoir, déployer et maintenir des systèmes modernes centrés sur la flexibilité, tout en capitalisant sur les enseignements liés à la conteneurisation et à la cybersécurité.

Prise en compte des dimensions socio-environnementales



Prérequis

Connaissances conteneurisation et réseaux

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	
Cours intégré (cours + TD)	14
TD	
TP	15
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	1
Travail personnel	6

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

Mettre en place une application a base de micro-services.

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

Alternance 1h30 de cours, 1h30 de TP.
TP interactif avec rédaction d'une documentation et remise de livrables à la fin de la formation

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Contrôle, TP avec rapport
Copies corrigés et consultables sur demande

Plan de cours

Deprecated: htmlspecialchars(): Passing null to parameter #1 (\$string) of type string is deprecated in **C:\Developpement\syllabus\public_html\views\syllabus_template.php** on line **292**

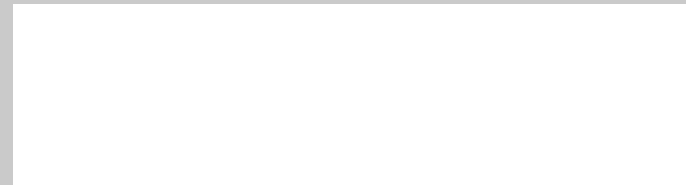
Ressources et références

Les supports pédagogiques sont disponibles en ligne sous Campus

Contexte et enjeux de l'enseignement

L'industrialisation du développement logiciel repose aujourd'hui sur des pratiques robustes d'intégration et de déploiement continu (CI/CD), essentielles pour garantir la qualité, la rapidité et la fiabilité des livraisons. Combinée à la conteneurisation, cette approche permet de maîtriser les environnements, d'automatiser les tests et les mises en production, tout en assurant une haute disponibilité et une scalabilité des applications. La maîtrise de ces outils et méthodes est devenue incontournable pour tout ingénieur impliqué dans le cycle de vie d'un logiciel moderne. Cet enseignement vise à fournir une compréhension opérationnelle des chaînes CI/CD et des technologies de container comme Docker, en s'appuyant sur des cas concrets et des outils standard de l'industrie.

Prise en compte des dimensions socio-environnementales



Prérequis

Infrastructure as Code Containérisation

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	
Cours intégré (cours + TD)	19
TD	
TP	
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	1
Travail personnel	8

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

- Concevoir une chaîne d'intégration et de déploiement continu.
- Automatiser les tests et le déploiement d'une application ou d'un script
- Utiliser Docker pour créer et gérer des conteneurs.
- Structurer des pipelines CI/CD avec des outils comme GitLab CI ou Jenkins.
- Intégrer les pratiques DevOps dans un cycle de développement logiciel ou de configuration système

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

Cours
TP de mise en place de la chaine CI/CD

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

QCM
TP évalué

Plan de cours

- Introduction à l'intégration et au déploiement continu
- Historique et principes du CI/CD
- Avantages, limites et retours d'expérience industriels
- Panorama des outils du marché
- Automatisation des tests et gestion des versions
- Tests unitaires, d'intégration et de non-régression
- Couverture de code et gestion des dépendances
- Intégration continue via Git, hooks et outils associés
- Mise en œuvre d'une chaîne CI/CD complète
- Création de pipelines avec GitLab CI ou Jenkins
- Déploiement automatique sur un environnement de test
- Gestion des secrets, des variables d'environnement et de la sécurité

Ressources et références

Les supports pédagogiques sont disponibles en ligne sous Campus