

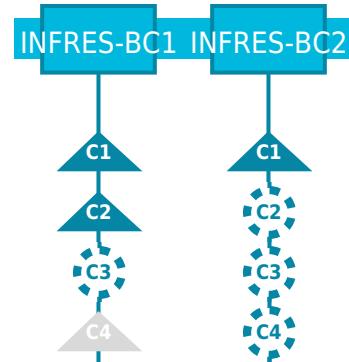
Pourquoi cette UE ?

Ce module associe gestion de parc informatique et microservices pour offrir aux administrateurs systèmes une vision complète des environnements modernes. La gestion de parc leur apprend à maîtriser et sécuriser l'ensemble des équipements et ressources, tandis que les microservices introduisent des architectures distribuées et scalables, désormais courantes en production. Ensemble, ces cours préparent à administrer efficacement des infrastructures hybrides, mêlant postes de travail, serveurs et applications cloud.

Eléments constitutifs de l'UE

| coefficients | | |
|--|--------------------------------------|---------------|
| | 1 | 1 |
| Volume d'heures d'enseignement encadré | Volume d'heures de travail personnel | Nombre d'ECTS |
| 38 | 0 | 2 |

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

La gestion de parc informatique est un enjeu stratégique pour toute organisation numérique. Elle englobe l'inventaire, le déploiement, la supervision, la maintenance et la sécurisation des équipements et logiciels utilisés par les utilisateurs. Dans un contexte d'industrialisation des systèmes d'information, de télétravail croissant et de menaces informatiques, une gestion efficace permet de garantir disponibilité, performance et sécurité du système d'information tout en maîtrisant les coûts. Cet enseignement vise à fournir aux élèves les compétences pour analyser, concevoir et administrer un parc informatique de manière professionnelle, évolutive et responsable.

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

Prérequis

Administration système Administration réseau

Modalités d'enseignement et d'évaluation

| | Nb d'heures |
|------------------------------|-------------|
| Cours | |
| Cours intégré (cours + TD) | 17 |
| TD | |
| TP | |
| Projets | |
| Travail en autonomie encadré | |
| Contrôles et soutenances | 1 |
| Travail personnel | |

Objectifs pédagogiques

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

- Recenser et inventorier les éléments d'un parc informatique.
- Déployer et configurer des postes clients à l'échelle d'une organisation.
- Mettre en œuvre une supervision et une maintenance proactive.
- Sécuriser les postes et gérer les politiques d'accès.
- Évaluer les besoins d'évolution d'un parc en fonction des usages.

Activités

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

Cours
TP**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 la gestion de parc informatique : typologie des équipements, enjeux métiers, cycle de vie d'un poste de travail, outils d'inventaire, normalisation des configurations.
- Déploiement automatisé de systèmes et logiciels : gestion d'images système, serveurs de déploiement, scripts d'automatisation, gestion de licences et mises à jour logicielles.
- Supervision et maintenance : outils de monitoring (Zabbix, GLPI, etc.), gestion des incidents, tableaux de bord, processus ITIL liés à la gestion d'un parc.
- Sécurité et conformité : politiques de sécurité des postes clients, contrôle des accès, chiffrement, protection contre les malwares, RGPD et traçabilité.
- Évolutivité, rationalisation et budget : gestion des cycles de renouvellement, intégration au système d'information, analyse de coûts, green IT et choix technologiques durables.

Ressources et références

Support de cours disponible sur Campus.

Contexte et enjeux de l'enseignement

Face à l'essor des architectures distribuées, les administrateurs système sont de plus en plus confrontés à des environnements composés de microservices, où la frontière entre exploitation et développement devient floue. Comprendre le développement de microservices permet d'adapter les outils d'exploitation, d'automatiser les tâches complexes, de concevoir des composants techniques (probes, agents, API internes) et d'optimiser la supervision. Cet enseignement vise à doter les élèves d'une capacité à développer des microservices simples, robustes et maintenables, spécifiquement pensés pour améliorer l'efficacité, la résilience et la sécurité des systèmes en production. L'enjeu est de donner aux futurs administrateurs système une autonomie accrue dans les environnements DevOps et cloud-native.

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

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 | |

Objectifs pédagogiques

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

- Concevoir un microservice adapté aux besoins d'un système d'exploitation ou d'administration
- Développer un service simple, scriptable, observable et sécurisé
- Intégrer des microservices dans des workflows d'automatisation ou de monitoring
- Maintenir la traçabilité, la journalisation et les métriques des services développés

Activités

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

- Cours
TP

Évaluations et retours faits aux élèves

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

- TP évalué

Plan de cours

- Introduction : rôle du microservice dans l'administration système
Cas d'usage : sonde personnalisée, interface d'administration, proxy local, gestionnaire d'état
- Notions de service stateless, résilience, redémarrage automatique
Avantages du développement ciblé par rapport à l'utilisation exclusive d'outils externes
- Bases du développement de microservices
Choix du langage (Go, Python, Bash avancé) adapté à l'environnement système
- Structure d'un microservice : points d'entrée, interfaces simples (REST, gRPC, CLI)
Gestion des erreurs, logs, métriques, configuration externe
- Conteneurisation et intégration dans l'environnement système
Création d'images minimales et sûres (Docker, Podman)
- Intégration avec systemd, cron, sockets Unix, volumes partagés
Communication inter-service (localhost, DNS interne, service discovery)

Ressources et références

Support de cours disponible sur Campus.