

Pourquoi cette UE ?

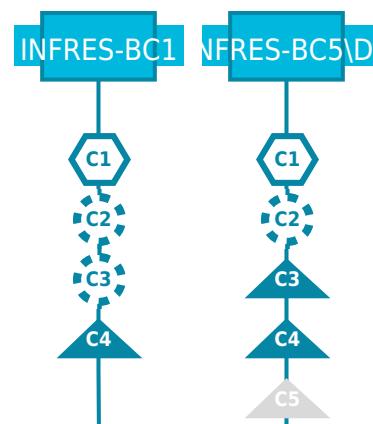
Ce module forme les étudiants aux fondements techniques indispensables en environnement DevOps. Il combine la maîtrise du langage de script Python, essentiel pour l'automatisation et les outils d'intégration, avec les compétences en administration système Linux, socle de la majorité des infrastructures actuelles. Le cours d'Infrastructure as Code complète l'ensemble en introduisant les bonnes pratiques de gestion déclarative et reproductible des systèmes. Ce module développe ainsi l'autonomie, la rigueur et la maîtrise des environnements modernes d'exploitation.

Eléments constitutifs de l'UE

	coefficients	
INFRES_8_1_SR-1 Langage de script	3	
INFRES_8_1_SR-2 Administration système Linux	3	
INFRES_8_1_SR-3 Infrastructure as code	2	
Volume d'heures d'enseignement encadré	Volume d'heures de travail personnel	Nombre d'ECTS
87	30	4

Alignement curriculaire

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



Contexte et enjeux de l'enseignement

Les langages de script sont aujourd'hui au cœur des outils d'ingénierie, d'automatisation et de prototypage rapide. Python, en particulier, s'est imposé comme un langage incontournable grâce à sa simplicité, sa lisibilité et sa vaste écosystème de bibliothèques. Cet enseignement s'inscrit dans un contexte où les ingénieurs doivent être capables de concevoir rapidement des outils efficaces pour traiter des données, automatiser des tâches techniques ou interagir avec des systèmes complexes. La maîtrise de Python offre une base solide pour aborder l'analyse numérique, le développement d'applications scientifiques, le traitement de fichiers, ou encore le pilotage d'équipements et la création de scripts d'intégration logicielle. Ce cours constitue ainsi un socle de compétences fondamentales et immédiatement mobilisables dans tout environnement technique ou scientifique.

Prise en compte des dimensions socio-environnementales**Prérequis**

L'apprenti doit connaître le système et le réseau, et adopter la manière « pythonique » de programmer, en utilisant un haut niveau, contrairement à C ou Java.

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	10

Objectifs pédagogiques

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

- Savoir installer Python, iPython, les modules Python,
- Savoir organiser un programme,
- Savoir programmer pour le système,
- Savoir programmer pour le réseau.

Activités

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

Le cours sera projeté en utilisant iPython notebook (l'ensemble des textes et exemples seront préparés à l'avance). Des exercices d'application réguliers seront proposés et des exercices plus complexes et plus généraux seront donnés en fin de journée et plus particulièrement en fin de formation. Les cours et TP alterneront.

Évaluations et retours faits aux élèves

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

Projet et Contrôle
Copies corrigés et consultables sur demande

Plan de cours

- Présentation du langage (syntaxe, types),
- Présentation des particularités du haut niveau,
- Présentation des trois paradigmes essentiels (procédural, objet et fonctionnel),
- Programmation système (entrées, sorties, appels de programmes externes, écriture d'un parseur d'arguments),
- Programmation réseau (interfaces de bas niveau et de haut niveau à la fois pour UDP et TCP, création d'un serveur et client HTML).

Ressources et références

Les supports pédagogiques sont disponibles en ligne sous Campus.

Contexte et enjeux de l'enseignement

L'administration avancée de systèmes Linux est un enjeu central pour tout ingénieur amené à concevoir, déployer ou maintenir des infrastructures informatiques robustes et sécurisées. Au cœur des environnements de production, des serveurs aux systèmes embarqués, Linux s'impose par sa stabilité, sa flexibilité et sa large adoption dans l'industrie. Ce cours vise à doter les élèves des compétences nécessaires pour maîtriser la configuration fine du système, l'automatisation, la sécurité, la supervision, ainsi que la gestion avancée des réseaux et des services. Il s'inscrit dans un contexte où la maîtrise de l'écosystème Linux est un atout stratégique pour répondre aux besoins croissants en performance, en résilience et en cybersécurité des infrastructures modernes.

Prise en compte des dimensions socio-environnementales

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	
Cours intégré (cours + TD)	12
TD	
TP	16
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	2
Travail personnel	10

Prérequis

Connaissances de base en administration système.

Objectifs pédagogiques

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

- Configurer et administrer un système Linux en ligne de commande.
- Automatiser des tâches avec des scripts shell et des outils comme Ansible.
- Gérer la sécurité du système et des services réseau.
- Superviser les ressources et diagnostiquer des dysfonctionnements.
- Déployer et maintenir des services critiques en environnement multi-utilisateur.

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

- Maîtrise de l'environnement système : gestion avancée des utilisateurs et permissions, processus, services (systemd), configuration du noyau, gestion des journaux (journald, rsyslog).
- Automatisation et scripting : écriture de scripts shell robustes, utilisation d'outils d'automatisation (cron, systemd timers, Ansible), gestion des tâches planifiées et centralisation de la configuration.
- Réseau et sécurité : configuration réseau avancée, firewall (iptables/nftables), VPN, gestion des accès SSH, sécurisation des services, contrôle d'intégrité (auditd, SELinux/AppArmor).
- Supervision et diagnostic : outils de supervision (top, htop, iostat, netstat, nmap, Prometheus, Grafana), analyse de logs, gestion des alertes et dépannage système.
- Services et déploiement : mise en place de services (DNS, DHCP, HTTP, SMTP, NFS), virtualisation (KVM, containers LXC/LXD), sauvegarde/restauration, haute disponibilité et bonnes pratiques de maintenance.

Ressources et références

Les supports pédagogiques sont disponibles en ligne sous Campus

Contexte et enjeux de l'enseignement

L'infrastructure as code (IaC) transforme la gestion des systèmes d'information en la rendant programmable, reproductible et versionnable. Dans un contexte d'automatisation croissante des déploiements, les infrastructures doivent être définies, modifiées et déployées de manière fiable, rapide et sécurisée. IaC permet de réduire les erreurs humaines, d'assurer la cohérence entre environnements et de favoriser l'agilité des équipes techniques. Cet enseignement vise à doter l'élève ingénieur des fondements méthodologiques et techniques nécessaires pour concevoir, maintenir et sécuriser des infrastructures automatisées, au service de systèmes complexes et distribués.

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

Prérequis

Virtualisation Administration Linux

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	
Cours intégré (cours + TD)	27
TD	
TP	
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	
Travail personnel	10

Objectifs pédagogiques

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

- configurer un parc de machines virtuelles à l'aide de l'outil Ansible,
- Assimiler les notions fondamentales de cet outil : control host, targets, commandes ad hoc, modules Ansible, idempotence, playbooks, facts, rôles, etc.

Activités

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

- TD
- Installation d'un cluster de machines virtuelles
- Exposé théorique des notions
- Nombreux ateliers pratiques pour l'assimilation par la pratique
- Exercices pour chacune des notions

Évaluations et retours faits aux élèves

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

Exercices en cours (contrôle continu)

Plan de cours

Présentation d'Ansible
Installer Ansible sur le Control Host
Configuration de base
Les commandes Ad-hoc
C'est quoi l'idempotence ?
Facts et variables implicites
Cibles hétérogènes
Jinja & Templates

Ressources et références

<https://blog.microlinux.fr/formation-ansible/>