

IDENTIFICATION

Course code : ESI-SPI-CI-IN4-
S7-UE1-EC2
ECTS : 1

DURATION

Lectures : 4
Tutorial classes : 10
Labs : 6
Total : 20

Project :
Personnal work :

ASSESSMENT

At least two continuous assessments.

TEACHING MATERIALS

Slides, practical work sheets.

LANGUAGE

English

CONTACT

Master modern tools and practices for deploying, orchestrating, and automating information systems. This course trains students in containerizing applications with Docker, orchestrating them at scale with Kubernetes, and automating system administration with Ansible. It also covers essential infrastructure services for deploying an IS (LDAP directory, DNS, messaging) and cloud paradigms, enabling students to design, deploy, and maintain a complete information system while respecting security and availability constraints.

UE : UE7-SYSRES

COURSE : System Administration : Orchestration, Containers and Automation

SCOPE OF THE COURSE

OBJECTIVES

COMPETENCE AND SKILLS

- Design, deploy, and maintain essential information system infrastructure services (directory, DNS, messaging) while respecting security constraints
- Containerize an application using Docker and manage its lifecycle : building images, composing multi-container services, image registries
- Deploy and administer a Kubernetes cluster : managing workloads, services, persistent volumes, scaling, and continuous updates
- Automate system infrastructure configuration and deployment using Ansible : playbooks, roles, dynamic inventories
- Design, deploy, maintain, and evolve cloud services by distinguishing IaaS, PaaS, and SaaS models, and evaluating the advantages and limitations of each approach
- Implement data collection and analysis mechanisms for preventive maintenance and anomaly detection in the information system

PROGRAM

- **Information system infrastructure services :**
 - Authentication directory : LDAP protocol — structure, schema, deployment, and integration into an IS.
 - DNS hierarchy : zones, records, recursive and authoritative resolution.
 - Installing and securing a DNS server : DNSSEC, signed zone management.
 - Configuring a mail server : SMTP, IMAP protocols, anti-spam, DKIM, SPF, DMARC.
 - Centralized identity management : LDAP integration with IS services (DNS, mail, SSH).
- **Containerization with Docker :**
 - Recap of virtualization and positioning of containers : namespaces, cgroups, image layers.
 - Building Docker images : Dockerfile, best practices, layer optimization, multi-stage builds.
 - Managing the lifecycle of containers : Docker networks, persistent volumes, environment variables.
 - Composing services with Docker Compose : defining multi-container stacks, dependencies, health checks.
 - Image registries : Docker Hub, private registry (Harbor); versioning and security policies.
 - Container security : non-root user, vulnerability scanning (Trivy), signed content policy.
- **Orchestration with Kubernetes :**

- Kubernetes architecture : control plane (API server, etcd, scheduler, controller manager), worker nodes.
- Core objects : Pod, Deployment, ReplicaSet, Service, ConfigMap, Secret, Namespace.
- Managing persistent storage : PersistentVolume, PersistentVolumeClaim, StorageClass.
- Scaling and continuous updates : *rolling update, rollback, Horizontal Pod Autoscaler*.
- Exposing services : Ingress, LoadBalancer, TLS certificate management (cert-manager).
- Security in Kubernetes : RBAC, NetworkPolicies, Pod Security Standards, secret management.
- Deployment tools : Helm (Kubernetes package manager) ; introduction to GitOps (ArgoCD).
- **Automation with Ansible :**
 - Automation principles : idempotence, infrastructure as code, configuration management.
 - Ansible architecture : inventories (static and dynamic), modules, playbooks, variables, and facts.
 - Structuring Ansible projects : roles, collections, Ansible Galaxy.
 - Automating administration tasks : service deployment, user management, updates.
 - Integrating Ansible into a CI/CD pipeline : automatic triggering, secret management (Ansible Vault).
- **Cloud deployment of containerized applications :**
 - Cloud-native vs internal applications : design differences, scalability, resilience.
 - Deploying a containerized application on a managed Kubernetes service (AKS, GKE, or EKS).
 - DevOps and CI/CD principles : continuous integration and deployment pipeline (GitLab CI, GitHub Actions).
 - *IaaS/PaaS/SaaS paradigms and cloud architecture choices are covered in depth in SR802.*
- **Observability in a containerized environment (introduction) :**
 - Observability challenges in Kubernetes : metrics, logs, traces.
 - Application and system metrics : exposure via Prometheus, visualization with Grafana — basic concepts.
 - Logging in Kubernetes : stdout/stderr, log aggregation with Loki.
 - *These tools will be covered in depth in SR802 — System Administration : Storage and Monitoring.*

BIBLIOGRAPHY

- Evi Nemeth et al. - *UNIX and Linux System Administration Handbook* - Addison-Wesley, 5th ed., 2017

- Kelsey Hightower et al. - *Kubernetes : Up and Running* - O'Reilly, 3rd ed., 2022
- Jeff Nickoloff & Stephen Kuenzli - *Docker in Action* - Manning, 2nd ed., 2019
- Lorin Hochstein - *Ansible : Up and Running* - O'Reilly, 3rd ed., 2022
- Kubernetes Documentation : <https://kubernetes.io/docs/>
- Docker Documentation : <https://docs.docker.com>
- Ansible Documentation : <https://docs.ansible.com>
- Helm Documentation : <https://helm.sh/docs/>
- ANSSI - *Security recommendations for Docker container deployment* : <https://www.ssi.gouv.fr/guide/>

REQUIREMENTS

- **SR051 - Networking and IoT Principles (S5)** : OSI model, TCP/IP, routing, DNS — essential for understanding Docker and Kubernetes networks.
- **SR052 - Operating System and Command Language (S5)** : Linux administration, process management, permissions, Bash scripting — foundation for all system administration.
- **SR062 - System Administration 1 (S6)** : system services (systemd), user management, introduction to network services — direct prerequisite for LDAP, DNS, and messaging.
- **SR603 - Virtual Environments and Collaboration Practices (S6)** : introduction to containers, collaboration tools, version control — foundations of containerization and GitOps.
- **Expected cross-disciplinary skills :**
 - Common Linux administration : service management, networking, permissions, text editors (vim/nano)
 - Basic Bash scripting : loops, conditions, file management
 - Git proficiency : clone, commit, push, branches
 - Basic knowledge of YAML and JSON for DevOps tool configuration
 - Ability to read technical documentation in English