SYLLABUS

DevOps: Cloud Automation

University year 2026-2027

1. Information regarding the programme

1.1. Higher education institution	Universitatea Babeș-Bolyai Cluj-Napoca
1.2. Faculty	Facultatea de Matematică și Informatică
1.3. Department	Departamentul de Informatică
1.4. Field of study	Informatică
1.5. Study cycle	Licență
1.6. Study programme/Qualification	Informatică in limba engleză
1.7. Form of education	Cu frecvență

2. Information regarding the discipline

2.1. Name of the discipline	**DevOps: Cloud Automation**					Discipline code	**MLE5267**
2.2. Course coordinator Dr. Catalin Rusu							
2.3. Seminar coordinator Dr. Catalin Rusu							
2.4. Year of study	3	2.5. Semester	5	2.6. Type of evaluation	C	2.7. Discipline regime	DO

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	**5**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**1L+2P**
3.4. Total hours in the curriculum	70	of which: 3.5 course	28	3.6 seminar/laboratory/project	**42**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					8
Additional documentation (in libraries, on electronic platforms, field documentation)					8
Preparation for seminars/labs, homework, papers, portfolios and essays (mai mare sau egal cu nr. total ore prevăzut în calendarul disciplinei pentru temele de control)					8
Tutorship					4
Evaluations					2
Other activities: [de ex.: comunicare bidirecțională cu titularul de disciplină / tutorele]					
3.7. Total individual study hours	**30**				
3.8. Total hours per semester	**100**				
3.9. Number of ECTS credits	**4**				

4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	

5. Conditions (if necessary)

5.1. for the course	• Beamer; • Access to Cloud Platforms.
5.2. for the seminar /lab activities	Knowlege of at least a programming language.

6.1. Specific competencies acquired [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	• Advanced capability to analyze, design, and securely build information systems, using a diverse range of hardware and software platforms, programming languages and environments, and modeling, verification, and validation tools; • The ability to develop distributed Cloud applications, in compliance with ethical and security requirements; • Acquiring a solid theoretical and practical foundation regarding communication over insecure media, as well as the use of secure Internet protocols.
Transversal competencies	• Professional communication skills: clear, concise description of professional results, both verbal and written; • Honorable, ethical conduct, adherence to professional deontology; • Initiative and entrepreneurial spirit; working knowledge of economics, continuous learning.

6.2. Learning outcomes

Knowledge	**The student knows:** • the fundamental concepts of containerization (Docker) and container orchestration (Kubernetes); • the architecture of a Kubernetes cluster, including service types, ephemeral and persistent volumes (PV, PVC, StorageClass), ConfigMaps, Secrets, Ingress, and readiness probes; • the principles of Infrastructure as Code and cloud-native application design; • the security challenges specific to distributed cloud applications and the mechanisms used to address them.
Skills	**The student is able to:** • containerize applications using Docker and manage container images and lifecycles; • deploy, configure, and manage workloads on a Kubernetes cluster using YAML manifests; • organize and isolate cluster resources using namespaces, labels, and annotations; • provision and manage cloud infrastructure using Infrastructure as Code practices on a public cloud provider (Azure); • apply security best practices when designing and deploying cloud-native applications.
Responsibility and autonomy:	The student has the ability to work independently to obtain a fully deployed, containerized, cloud-native application, from initial design through deployment and configuration on a public cloud platform, while applying appropriate security and quality standards.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Introducing students to cloud computing concepts, along with the challenges involved in developing cloud-native applications • Introducing students to Infrastructure as Code concepts • Presenting real use cases and the benefits of using a public cloud provider • Presenting security topics
7.2 Specific objective of the discipline	• Familiarization with Azure • Practical experience in cloud application architecture

8. Content

8.1 Course	Teaching methods	Remarks
1 – 2 Docker Basics	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
3 – 5 Docker Advanced	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
6 Kubernetes	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
7 – 8 Service types, Ephemeral Volumes	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
9 YAML Basics	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
10 PV, PVC, Storage Class	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
11 ConfigMaps & Secrets	Expunere: descriere, explicații, exemple practice, demonstrații, discuții pe studii de caz.	
12 Ingress, Readiness probes	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
13 Namespaces, Labels, Annotations	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
14 Recapitulare si inchidere	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
Bibliography		
8.2 Seminar / laboratory	Teaching methods	Remarks
During labs, illustrative demonstrations and relevant examples on the course topic are given first. Then, students — individually or in teams — will solve problems assigned as lab exercises or as homework.	Lecture: description, explanations, practical examples, demonstrations, case study discussions.	
Bibliography		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

This course is delivered in partnership with msg systems Romania srl, to ensure quality, relevance, and access to industry-standard cloud resources.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	Knowledge of the technologies presented in the course	Colocviu	50%
10.5 Seminar/laboratory	Ability to implement the concepts learned	Test	50%
10.6 Minimum standard of performance			
•			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date:
10.09.2026

Signature of course coordinator

Dr. Catalin Rusu

Signature of seminar coordinator

Dr. Catalin Rusu

Date of approval:
April 2026

Signature of the head of department

Assoc.prof.phd. Adrian STERCA

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*".