Tallinn University of Technology
School of Information Technologies

ITI8801 Cloud Architectures and DevOps
Extended Syllabus
Autumn 2026

Course objectives:

The aim of the course is to:
- connect theoretical knowledge of distributed systems with the practical implementation of cloud-based architectures in production-like environments;
- develop the ability to design, implement and operate scalable, resilient and cost-efficient cloud infrastructures using modern DevOps practices;
- develop systematic understanding of infrastructure as code, continuous integration and deployment, container orchestration, observability and operational reliability;
- develop the ability to analyse and justify architectural and technical choices in cloud environments;
- develop collaboration, peer review and technical communication skills through code reviews, team-based development and technical presentations;
- provide practical experience with widely used cloud platforms, tools and

Learning outcomes:

After completing the course, the student:
1. designs and implements reliable, scalable and cost-efficient cloud architectures on major cloud platforms, applying principles of high availability, fault tolerance and security within defined cost constraints;
2. creates a comprehensive monitoring, logging and alerting solution using industry-standard tools, including Prometheus and Grafana, and analyses operational data to identify and resolve system issues;
3. applies DevOps practices throughout the software development lifecycle and implements automated CI/CD pipelines covering build, testing, security scanning and deployment to multiple environments;
4. manages cloud infrastructure programmatically using Infrastructure as Code principles and tools such as Terraform, ensuring reproducibility, version control and maintainability of infrastructure;
5. deploys and operates containerised applications using Docker and Kubernetes, including load balancing, Ingress, scaling, service mesh concepts and stateful workloads;
6. collaborates effectively in technical teams, performs structured code reviews, presents technical solutions and provides constructive feedback on architectural and implementation choices.

Short description of course content (topics):	The course provides practical experience in designing, deploying and operating cloud-based distributed systems. The course progresses from basic cloud infrastructure towards more complex production-style architectures and DevOps workflows. Main topics: 1. Cloud computing concepts and service models: IaaS, PaaS, serverless and SaaS; cloud regions, availability zones and the shared responsibility model. 2. Cloud networking, load balancing and identity management. 3. Infrastructure as Code and Terraform: state, modules and reproducibility. 4. Containers, Docker, container registries and security scanning. 5. Kubernetes: cluster architecture, deployments, services, Ingress and scaling; code reviews. 6. CI/CD using GitLab CI: build, test, scan and deployment to multiple environments. 7. Microservices architecture. 8. Observability using Prometheus, Grafana, centralised logging and alerting; incident exercise. 9. Service mesh and stateful workloads. 10. Zero-downtime deployment and disaster recovery. 11. Cloud cost management and FinOps, secrets management and cloud security.
Language of	English.
ECTS credits:	6 ECTS. The estimated total student workload is approximately 156 hours.
Students:	The course is primarily for students in groups IAIM11 and IAIM12. The course is delivered as an on-site course with lectures, practice sessions, independent assignments and a team project. Remote participation is supported on a best-effort basis where technically and pedagogically feasible. Remote participation cannot be guaranteed, and some activities may require live participation. The examination takes place on site.
Special needs:	Students with special educational needs are welcome to participate in the course. Students are encouraged to inform the lecturer as early as possible about any special needs or reasonable adjustments that may affect participation in learning or assessment activities so that appropriate arrangements can be
Registration:	Students wishing to take the course must declare the course in ÕIS in accordance with the deadlines specified in the academic calendar.
Prerequisite courses and/or required prior knowledge:	There are no formal prerequisite courses specified for the course. Students are expected to have the following practical prior knowledge: - ability to write a small application in a programming language of their choice; - basic understanding of REST and JSON; - basic Git skills, including clone, branch, commit and push; - basic Linux command-line skills; - understanding of the concepts of an IP address and a network port. Prior knowledge of cloud platforms, Docker, Kubernetes, Terraform or monitoring systems is not required. These are topics taught during the course.

Required resources:

https://confluence.ttu.ee/x/Qo9aAQStudents need a computer with an internet connection, Git and access to the course repository, a personal cloud account, and the development and command-line tools introduced during the course.

Practical work is completed using the student's own cloud account. The default platform is Azure for Students, which is available free of charge and does not require a credit card. AWS Free Plan may be used as an alternative. All mandatory individual assignments are designed to be completable on either supported platform.

Version control is performed using Git. The student's course repository is named ITI8801-2026.

Students are responsible for managing their own cloud resources and

Lecturer:

Siim Rebane, siim.rebane@taltech.ee

Contacting the lecturer:

The lecturer can be contacted by e-mail, through Microsoft Teams, and during consultation sessions. Consultations are available on request. Scheduled consultation times are published on Moodle.

Contact study schedule:

Teaching takes place during weeks 1–16.

Lecture: Monday, 14:00–15:30, MEK-114, groups IAIM11 and IAIM12.

Practice session for IAIM11: Monday, 16:00–17:30, MEK-110.

Practice session for IAIM12: Wednesday, 16:00–17:30, MEK-126.

Lectures and practical sessions are recorded using Echo360 where possible.

Description of the learning process:

The total workload of the 6 ECTS course is approximately 156 hours.

Approximate workload distribution:
- contact study: 64 hours (16 × 2 h lecture + 2 h practice session);
- independent study: approximately 92 hours.

Contact study consists of lectures and practical sessions throughout the 16-week semester. Lectures introduce the conceptual, technical and architectural foundations of each topic. Practical sessions focus on applying the concepts in real cloud environments, working with infrastructure, automation, containers, Kubernetes, observability tools and deployment pipelines.

Independent work includes completing ten individual practical assignments, configuring and operating cloud resources, maintaining the course Git repository, developing the team project, preparing for the technical presentation, reviewing course materials and preparing for the written examination.

The individual practical assignments primarily support and assess Learning Outcomes 1–5. The team project, code review activities and technical

E-learning support:	https://moodle.taltech.ee/course/view.php?id=34749The primary electronic learning environment for the course is Moodle. Moodle is used for course materials, announcements, individual assignments, deadlines, grades and feedback, consultation information, examination preparation materials and sample examination questions. Lectures and practice sessions are recorded in Echo360 where possible. Lecture slides are also available at: https://iti8801.digivader.com
Literature:	Main literature: Newman, S. (2021). Building Microservices: Designing Fine-Grained Systems. 2nd edition. O'Reilly Media. Morris, K. (2020). Infrastructure as Code: Dynamic Systems for the Cloud Age. 2nd edition. O'Reilly Media. Additional sources include official Microsoft Azure, AWS, Kubernetes and Terraform documentation, as well as other technical documentation and reading materials assigned during the course. There is no mandatory advance reading. Specific reading materials are assigned during the course.
Assessment	Examination.

Continuous assessment, assessment methods, criteria and deadlines.

The course is graded on a 100-point scale.

Final result:
- individual assignments: 20 points / 20%;
- team project and presentation: 20 points / 20%;
- written examination: 60 points / 60%.

In addition to the total score, a separate examination threshold applies. To pass the course, the student must obtain at least 30 out of 60 points, or 50%, on the written examination. All ten individual assignments must be submitted before the student is eligible to take the examination.

Individual practical assignments – 20 points
During the course, students complete 10 individual practical assignments. Each assignment is worth a maximum of 2 points. There are 10 assignments in total; each is published on a Monday via Moodle. Each assignment requires the student to produce one concrete and verifiable technical result in their own cloud environment.

For example, an early assignment may require a service at https://<own-domain>/hello to return the student's code in JSON format. The solution may require a manually created virtual machine, a self-created DNS A record, a valid HTTPS certificate and a working deployed application. The student submits the required URL and a cloud-portal screenshot through Moodle.

Assessment:
- correctly completed and submitted by the deadline: 2 points;
- submitted up to one week after the deadline: 1 point;
- submitted more than one week after the deadline: 0 points.

Even when an assignment is too late to receive points, submission remains mandatory. Assignments are normally published on Monday together with the corresponding lecture. The deadline is Sunday of the following week at 23:59.

Relationship to learning outcomes: individual assignments assess Learning Outcomes 1–5 depending on the specific technical topic.

Team project and presentation – 20 points
During the second half of the semester, students form teams and design and implement a larger cloud-based system. The team project and presentation are assessed at the final presentation.

Assessment considers:
- the demonstrated working system;
- the architectural choices made by the team;
- the quality and clarity of the justification for those architectural choices;
- each team member's individual contribution.

The team must be able to demonstrate that the solution works and explain the main architectural and technical decisions. Each team member must be prepared to describe and justify their own contribution and answer questions related to their work.

Maximum result: 20 points. The detailed point breakdown and assessment

Examination:

The examination is a written, on-site examination. The examination is completed using pen and paper. No aids are permitted.

The examination is worth a maximum of 60 points and contributes 60% of the final grade. The minimum passing threshold for the examination is 30 out of 60 points, or 50%.

Prerequisite for taking the examination: all 10 individual assignments must have been submitted. An assignment submitted too late to receive points still fulfils the submission prerequisite. An assignment that has not been submitted does not fulfil the prerequisite.

Sample questions are published before the examination. Consultation sessions are organised before the examination. The exact examination date and time are published in ÕIS and Moodle.

Examination assessment criteria and points:

The written examination assesses the student's individual understanding of the main concepts, technologies and architectural principles covered during the course.

The examination may cover cloud service models, regions and availability zones, the shared responsibility model, networking and load balancing, identity and access management, Infrastructure as Code, Terraform concepts, containers and Docker, CI/CD, GitLab CI, Kubernetes architecture and operations, microservices, observability, monitoring, logging and alerting, service mesh, stateful workloads, zero-downtime deployment, disaster recovery, cloud security, secrets management, FinOps and cost management, cloud migration, and architectural analysis and trade-offs.

The student must demonstrate independent understanding and the ability to reason about technical and architectural decisions. Artificial intelligence tools,

Final grade:

The final result is calculated as follows:
- individual assignments: maximum 20 points;
- team project and presentation: maximum 20 points;
- written examination: maximum 60 points.

Total: maximum 100 points.

To receive a positive final grade, both conditions must be fulfilled:
1. the student receives at least 30 out of 60 points on the written examination;
2. the total course score is at least 51 out of 100 points.

The final score is converted into a grade as follows:
91–100 points = 5, excellent
81–90 points = 4, very good
71–80 points = 3, good
61–70 points = 2, satisfactory
51–60 points = 1, sufficient
0–50 points = 0, fail

If the examination threshold of 30 out of 60 points is not achieved, the course cannot be passed regardless of the total number of points collected from other assessment components.

Academic practices:

Head tavad | Õppeinfo | TalTechStudents of Tallinn University of Technology are required to follow the University's principles of good academic practice. Work submitted under a student's name must represent the student's own work or, in the case of explicitly assigned team work, the work of the declared team.

Plagiarism, cheating, submitting another person's work as one's own, falsifying results or evidence, or using AI tools contrary to the rules specified for an assessment activity are not permitted. Unauthorised assistance may result in a score of 0 for the relevant assignment or assessment and may lead to further proceedings under TalTech regulations.

Students must follow TalTech's good academic practice requirements when using generative AI. Where AI use is permitted, students remain responsible for the correctness and quality of the submitted result and must be able to explain and defend their work.

During the written examination, the use of AI tools, internet resources, unauthorised materials or assistance from another person is prohibited.

If remote assessment or examination monitoring involving screen recording or similar recording technology is ever used, this may only be done with the student's explicit written consent in a form that can be reproduced. Without such consent, recording and subsequent review of the student's screen or activity may not be performed. Where required under TalTech rules, an alternative assessment arrangement must be offered.

Detailed schedule and topics

The semester plan is preliminary and may be adjusted if necessary. Detailed instructions, assignment specifications and any schedule changes are published on Moodle.

Week 1 – Introduction to cloud computing

Topics: what cloud computing is; IaaS, PaaS, serverless and SaaS; regions and availability zones; shared responsibility model; introduction to the course; preparation of the cloud environment.

Week 2 – Networking, load balancing and identity management

Topics: cloud networking; IP addressing; DNS; security rules and firewalls; load balancing; identity and access management.

- **Week 3 – Infrastructure as Code I**
- Topics: Infrastructure as Code principles; Terraform; providers and resources; Terraform state; reproducible infrastructure.
    -

Week 4 – Infrastructure as Code II

Topics: Terraform modules; reuse and modularity; state management; version-controlled infrastructure; infrastructure lifecycle.

Week 5 – Containers

Topics: Docker; container images; container registries; image build process; container security; security scanning.

- **Week 6 – Kubernetes I**
- Topics: Kubernetes cluster architecture; Pods; Deployments; Services; application configuration.
-

Week 7 – CI/CD

Topics: continuous integration and continuous delivery; GitLab CI; automated build; automated testing; security scanning; deployment to multiple environments.

- **Week 8 – Kubernetes II and code reviews**
- Topics: Ingress; scaling; operating Kubernetes workloads; structured code reviews; technical peer feedback.

Week 9 – Microservices architecture

Topics: service boundaries; service communication; dependencies; resilience; architectural trade-offs in distributed systems.

Week 10 – Observability I

Topics: monitoring; metrics; Prometheus; Grafana; service and system health; performance monitoring.

Week 11 – Observability II and incident exercise

Topics: centralised logging; alerting; incident detection; incident analysis; troubleshooting; practical incident exercise.

Week 12 – Service mesh and stateful workloads

Topics: service-to-service communication; service mesh concepts; stateful applications; operating stateful workloads.
Team project development.

Week 13 – Zero-downtime deployment and disaster recovery

Topics: deployment strategies; zero-downtime deployment; rollback; high availability; backup and recovery; disaster recovery strategies.
Team project development.

Week 14 – Cost and security
Topics: FinOps; cloud cost monitoring; cost optimisation; secrets management; cloud security.
Team project development and finalisation.

Week 15 – Migrations
Topics: cloud migration strategies; migration risks; architectural trade-offs; examination preparation; discussion of sample examination questions.
Preparation of the team project presentation.

Week 16 – Team project presentations and course summary
Activities: demonstration of team solutions; technical presentations; justification of architectural choices; questions and discussion; demonstration of individual contribution; peer and lecturer feedback; course summary.
The detailed project assessment rubric is published on Moodle when the teams are formed during the second half of the semester.

-
-
-

-
-
-