GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	High Availability Distributed Systems, PG_00063936
Field of study	Informatics

Date of commencement of studies	February 2026	Academic year of realisation of subject	2025/2026
Education level	second-cycle studies	Subject group	Optional subject group Specialty subject group Subject group related to scientific research in the field of study
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	1	Language of instruction	English
Semester of study	1	ECTS credits	3.0
Learning profile	general academic profile	Assessment form	assessment

Conducting unit	Department of Computer Architecture -> Faculty of Electronics Telecommunications and Informatics -> Faculties of Gdańsk University of Technology

Name and surname of lecturer (lecturers)	Subject supervisor	dr Paweł Kozyra
	Teachers	dr Paweł Kozyra

Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	0.0	15.0	0.0	30
	E-learning hours included: 0.0						

Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
	Number of study hours	30	9.0	36.0	75

Subject objectives	The aim of the course is to introduce students to the subject of distributed applications development, as well as distributed data collection and processing systems. In addition, as part of the course, students will be familiarized with implementation platforms used in industry, whose task is to manage sets of distributed application components.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K7_W11] knows and understands, to an increased extent, the general principles of creation and development of forms of individual entrepreneurship and the economic, legal and other conditions of various types of activities related to the awarded qualification, including the principles of protection of industrial property and copyright law	The student can apply the principles of individual entrepreneurship	[SW1] Assessment of factual knowledge
	[K7_U12] is able, to an increased extent, to analyze the operation of components and systems related to the field of study, as well as to measure their parameters and study their technical characteristics, and to plan and carry out experiments related to the field of study, including computer simulations, interpret the obtained results and draw conclusions	The student can configure a parallel environment, run, and profile the execution of parallel applications	[SU1] Assessment of task fulfilment
	[K7_W10] knows and understands, to an increased extent, the basic processes occurring in the life cycle of equipment, objects and technical systems, as well as methods of supporting processes and functions, specific to the field of study	The student can apply knowledge about the life cycle of technical objects	[SW1] Assessment of factual knowledge
	[K7_W03] knows and understands, to an increased extent, the construction and operating principles of components and systems related to the field of study, including theories, methods and complex relationships between them and selected specific issues - appropriate for the curriculum	The student knows and describes various application development architectures. He knows the differences, advantages and disadvantages of using monolithic layered architectures and target distributed architectures.	[SW1] Assessment of factual knowledge

Subject contents	Course content – lecture Application scalability, deployment; Distributed application architectures (monolith -> micro services (CQRS / Event Sourcing / Saga); containerization of services -> docker, docker-compose, docker swarm, kubernetes, deployment and maintenance of a distributed application -> monitoring (clusters / clouds Computational OpenSTack / AWS) Monitoring -> Sentry / Jaeger / Promethes + Grafana / Load balancery / Queue systems; Locust.io / Jmeter load testing tools Distributed file systems HDFS (Hive) / IPFS Distributed databases (Hbase / Neo4j, ArangoDB) Blockchain - > Bitcoin / Ethereum / Stellar / GRP (graph) Distributed computing environment (Apache Spark / YARN -> JupyterLab -> PySpark -> .net context submit)
	Course content – project Defining the Kubernetes orchestrator. Performing operations with kubectl. Implementing and configuring multiple services. Deploying the full system locally. Setting up the Docker registry. Creating and auto-scaling the cluster. GitOps. Controlling the Kubernetes cluster. Configuring GitHub. Setting up multiple environments. Scaling the workflow. Setting up logs and metrics. Dealing with service dependencies.

Prerequisites and co-requisites	1. The student must have knowledge and programming skills in .net or java technologies. 2. The student must understand the methods of communication used on the Internet 3. The student must have knowledge and skills in implementing the application on the target server in the containerized version.

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	project	50.0%	50.0%
	lecture	50.0%	50.0%

Recommended reading	Basic literature	1. Cloud Native DevOps with Kubernetes, John Arundel, Justin Domingus. 2. Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications, Bilgin Ibryam, Roland Huß. 3. KUBERNETES: A Simple Guide to Master Kubernetes for Beginners and Advanced Users (2020 Edition), Brian Docker. 4. Hands-On Docker for Microservices with Python: Design, deploy, and operate a complex system with multiple microservices using Docker and Kubernetes, Jaime Buelta. 5. gRPC: Up and Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes, Kasun Indrasiri, Danesh Kuruppu.

	Supplementary literature	1. The Kubernetes Book, Nigel Poulton. 2. Hands-On Microservices with C# and .NET Core 3: Refactor you monolith architecture into microservices using Azure, 3rd Edition, Gaurav Aroraa, Ed Price. 3. Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages, Adam Freeman. 4. Practical Microservices Architectural Patterns - Event-Based Java Microservices with Spring Boot and Spring Cloud, Binildas Christudas. 5. Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith, Sam Newman. 6. Practical Microservices: Build Event-Driven Architectures with Event Sourcing and CQRS, Ethan Garofolo. 7. Architecting Modern Data Platforms, Jan Kunigk, Ian Buss, Paul Wilkinson & Lars George. 8. Advanced Analytics with Spark, Sandy Ryza, Uri Laserson, Sean Owen & Josh Wills. 9. Big Data Analytics with Hadoop 3, Sridhar Alla. 10. Modern Big Data Processing with Hadoop, V. Naresh Kumar Prashant Shindgikar.
	eResources addresses	
Example issues/ example questions/ tasks being completed	1. Explain what CNCF is 2. What is Infrastructure as a Code 3. RDD vs DataFrame 4. HDFS vs IPFS 5. pySpark vs python 6. The role of Yarn	
Practical activites within the subject	Not applicable	

Document generated electronically. Does not require a seal or signature.