# COURSE DESCRIPTION CARD - SYLLABUS

Course name
Introduction to cloud systems [S2Inf1-PB>SCHMUR]

## Course

Field of study
Computing

Year/Semester
1/1

Area of study (specialization)
Edge Computing

Profile of study
general academic

Level of study
second-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

## Number of hours

Lecture
30

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

## Number of credit points

4,00

## Coordinators

mgr inż. Damian Huderek
damian.huderek@put.poznan.pl

## Lecturers

## Prerequisites

The student starting the course should have basic knowledge of operating systems, network technologies, and security of distributed processing systems. Skills: The student should be able to obtain information from the indicated sources, as well as understand the need to expand his competences. Social Competences: The student should show such features as: honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

## Course objective

1. Provide students with basic knowledge of the functioning and construction of modern distributed cloud computing systems and issues related to the management and security of cloud systems. 2. Developing students" skills in solving technical problems in the field of complex digital system design. 3. Shaping teamwork skills in students - the ability to cooperate in the design teams and in the preparation of final research reports.

## Course-related learning outcomes

Knowledge:
1. has organized and theoretically grounded general knowledge related to key topics in the field of

computer science, the performance of selected solutions, consistency and correctness of selected algorithms (k2st_w2)
2. knows development trends and the most important new achievements in computer science and selected related scientific disciplines (k2st_w4)
3. knows the fundamental methods, techniques, and tools used to solve complex tasks in the selected area of computer science (k2st_w6)

Skills:
1. is able to acquire information from literature, databases and other sources (in the mother tongue and english), integrate them, interpret and critically evaluate them, draw conclusions and formulate and exhaustively justify opinions (k2st_u1)
2. can use analytical, simulation, and experimental methods to formulate and solve research problems (k2st_u4)
3. is able to combine knowledge from different areas of computer science (and if necessary from other scientific disciplines) to formulate and solve engineering tasks related to hardware-software development (k2st_u5)
5. can (e.g., by using new methods) solve complex it tasks with a research component (k2st_u10)
6. is able to design (according to a provided specification which includes also non-technical aspects) a digital system using technologies learned during the course (k2st_u11

Social competences:
1. understands that in computer science, knowledge and skills become obsolete very quickly (k2st_k1)
2. understands the importance of using the latest achievements in the field of computer science while solving research and practical problems (k2st_k2)
3. understands the importance of popularizing new achievements in the field of computer science (k2st_k3)

## Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
Formative assessment:
a) lectures: based on the answers to the questions which test understanding of material presented on the lectures
b) laboratory classes: based on the assessment of the tasks done during classes and as a homework
Summative assessment:
a) verification of assumed learning objectives related to lectures within an online written test.
b) verification of assumed learning objectives related to laboratories is carried out by assessing the tasks carried out during each laboratory meeting;
Getting extra points for activity during classes, especially for:
- proposing to discuss additional aspects of the issue,
- effectiveness of applying the acquired knowledge while solving a given problem,
- ability to work within a team that practically performs a specific task in a laboratory,
- comments related to the improvement of teaching materials.

## Programme content

The lecture program covers the following topics:
Introduction to containerization and microservices (Docker, Kubernetes). Architecture of cloud solutions. Managing data flow between services (Kubernetes CNI) and the structure and construction of connections in the cloud (Service Mesh). Telemetry and data flow analysis (TAS). Security of cloud solutions and data security. Acquainting with equipment dedicated to cloud solutions.
Laboratory classes are conducted in the form of 2-hour meetings, held in the laboratory, preceded by an instructional session at the beginning of the semester. The laboratory program covers the following topics:
Creating and using containers. Building network architecture. Management and supervision of the created architecture. Data flow visualization along with the analysis of this data. Network security analysis.
Part of the above-mentioned curriculum content is carried out as part of the student"s own work.

## Course topics

none

## Teaching methods

1. Lecture with multimedia presentation (diagrams, formulas, definitions, etc.) supplemented by the content of the board.
2. Laboratory exercises: multimedia presentation, presentation illustrated with examples given on the board and performance of tasks given by the teacher - practical exercises.

## Bibliography

Basic
1. Cloud Computing: Theory and Practice, D. Marinescu, Morgan Kaufmann 2013
2. J. Rosenberg, A. Mateos Chmura obliczeniowa. Rozwiązania dla biznesu, Helion, 2012
3. B. Sosinsky, Cloud Computing Bible, Wiley, 2010
4. N. Antonopoulos, L. Gillam, Cloud Computing: Principles, Systems and Applications, Springer, 2012
Additional
1. L. Wang, R. Ranjan, J. Chen, and B. Benatallah, Cloud Computing: Methodology, Systems, and Applications, CRC Press, Boca Raton

## Breakdown of average student's workload

| | Hours | ECTS |
|---|---|---|
| Total workload | 100 | 4,00 |
| Classes requiring direct contact with the teacher | 60 | 2,00 |
| Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation) | 40 | 2,00 |