UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Technological Infrastructures for Data Science

2526-2-FDS01Q016

Aims

Knowledge and Understanding

Upon completion of the course, students will have acquired:

- Foundational knowledge of the main technological solutions supporting data science, including virtualization, cloud computing, and containerization
- Understanding of reference architectural models for data science infrastructures, with particular focus on Data Lakes, HDFS, and YARN
- Theoretical knowledge of data processing paradigms (batch processing, stream processing, messaging) and related technologies (Hadoop, Spark, Storm, Kafka)
- Understanding of modern software development methodologies (Waterfall, Agile, DevOps, DataOps, MLOps) in the context of data science
- Basic knowledge to analyze and evaluate the possibility of applying existing infrastructural solutions to specific data science problems

Applying Knowledge and Understanding

Upon completion of the course, students will be able to:

- Design and implement containerization solutions using Docker for data science applications
- Use cloud tools and platforms for managing and processing large volumes of data
- Configure and use distributed systems for batch and stream data processing
- Apply appropriate software development methodologies to data science and machine learning projects
- Experiment with specific technical and technological solutions through practical laboratory activities
- Effectively interact with the technological tools presented during the course

Transversal Skills

The course contributes to the development of the following transversal skills:

Making Judgements: through critical analysis of different technological solutions and evaluation of their applicability to specific data science contexts.

Communication Skills: through oral discussion of in-depth topics from the course and the ability to present technical solutions in a clear and structured manner.

Learning Skills: by providing methodological tools to stay updated on the evolution of data science technologies and to independently explore new infrastructural solutions.

Contents

Topics

The course comprises the following modules:

Module 1 - Infrastructure: Introduction to Virtualization, Cloud Computing and Containerization.

Module 2 - Platform: Data organization and distribution, Data Lake, HDFS, YARN

Module 3 - Processing: Batch vs. Streaming vs. Messaging, the cases of Hadoop, Spark, Storm, Kafka

Module 4 - Software Development: Waterfall, Agile, DevOps, DataOps, MLOps

Detailed program

Course topics divided by modules:

Module 1 - Infrastructure

- The figure of the data engineer
- The reference architecture
- Virtualization
- Cloud Computing (Introduction, Service and deployment models, essential features)
- Containerization with Docker
- Serverless

Module 2 - Platform

- The Data Lake
- HDFS and YARN

Module 3 - Processing

- Batch processing (Apache Hadoop and Apache Spark).
- Stream processing (Apache Storm, Apache Spark, and Apache Flink)

- Messaging (Apache Kafka)

Module 4 - Software Development

- Service computing
- Software engineering
- Development methodologies (Waterfall, Agile, DevOps, DataOps, MLOps)

Prerequisites

Technical Fundamentals

- **Computer Systems Architecture**: understanding of hardware components organization and operation (processor, RAM memory, storage devices) and their interaction
- **Operating Systems**: familiarity with basic concepts and practical use of Unix/Linux or Windows environments
- **Command Line Interface**: proficiency in using shell/terminal for filesystem navigation, file management, and basic command execution

Programming Skills

- **Python Language**: knowledge of fundamental data structures (lists, dictionaries, tuples), control structures (loops, conditionals), functions, and basic library management
- **Jupyter Environment**: experience using Jupyter Notebook/Lab for interactive development, data visualization, and code documentation

Required Level

The listed competencies should correspond to those acquired in introductory computer science or programming courses at the undergraduate level, or through equivalent practical experience.

Teaching form

The course adopts an integrated teaching approach that combines different methodologies to promote active learning and student engagement:

Course Structure

Interactive Lectures

- Approximately 15 lectures of 2-3 hours each conducted in interactive mode in presence
- During lectures, the instructor will promote active student participation through questions, guided

discussions, case study analyses, and shared reflection moments
- Lectures will integrate theoretical content with practical examples and concrete applications

Lab Sessions

- 6 lab sessions of 3 hours each (or 9 lab sessions of 2 hours) conducted in presence with interactive methodology
- Hands-on activities for practical application of theoretical concepts
- Individual and group work under instructor supervision

Asynchronous Content

- A maximum of 10% of the course will be delivered asynchronously through video recordings
- Asynchronous content complements face-to-face lectures
- Students will be able to access this content according to their own study schedule to consolidate and expand on topics covered in class

Language of Instruction

The course will be entirely delivered in **English**

Textbook and teaching resource

Lecture notes and slide decks.

The following textbooks are referenced for further study:

- The basics of cloud computing ISBN-13: 978-0124059320 Authors: Derrick Rountree, Ileana Castrillo
- Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems ISBN-13: 978-1449373320 Author: Martin Kleppmann

Semester

Second year, first semester

Assessment method

The assessment of learning consists of **two complementary components**, both of which are mandatory:

Written Exam (25 points)

The written exam evaluates knowledge of the topics covered during the course through:

- **12–13 multiple-choice questions** to test understanding of fundamental concepts
- **2–3 open-ended questions** to assess analytical and synthesis skills

Duration: 60–75 minutes
Maximum score: 25 points

Oral Exam – In-Depth Presentation (7 points)

The oral exam involves the presentation and discussion of a topic chosen from:

- Subjects related to the course but not covered in class
- Further exploration into topics already discussed during lectures

Format:

- **Preparation**: research and slide creation may be carried out in groups of 2 people
- **Presentation and evaluation**: must be individual and personalized for each group member
- **Topic approval**: the topic must be approved in advance by the instructor

Maximum score: 7 points

Exam Passing Criteria

The exam is considered **passed** when **both** of the following conditions are met:

1. **Minimum threshold for each component**: score ? 15 in the written exam and ? 4 in the oral exam
2. **Total score**: the sum of the two components must be ? 18 points

Final grade: corresponds to the sum of the two scores (maximum 32 points, equivalent to 30 cum laude).

Important Notes

- **No partial exams** are scheduled during the course
- Failure to reach the minimum threshold in either component requires retaking **only that specific part**, not the entire exam

Office hours

Tuesday 12:30-14:30 ask for email confirmation

Sustainable Development Goals

INDUSTRY, INNOVATION AND INFRASTRUCTURE